A GENERAL PURPOSE THREE DIMENSIONAL STRESS ANALYSIS PROGRAM

Emmanuel Leonidas

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A GENERAL PURPOSE THREE DIMENSIONAL

STRESS ANALYSIS PROGRAM

by

Emmanuel Leonidas

Thesis Advisor: G. Cantin

December 1971

Approved for public release; distribution unlimited.

A General Purpose Three Dimensional Stress Analysis Program

by

Emmanuel Leonidas
Lieutenant Commander, Hellenic Navy

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

A computerized solution for three dimensional stress analysis was developed. The solution was based on the finite element technique employing twenty and thirty-two nodal point three dimensional isoparametric elements. The solution is applicable to linear elastic structures composed of homogeneous and isotropic materials subjected to static loading under constant temperature. Deformed states are restricted to small displacements and displacement gradients. The program is coded in standard FORTRAN IV and is machine independent except for one subroutine which may be adjusted to specific installations. Independence of core space requirements is achieved by use of direct access storage facilities.

TABLE OF CONTENTS

I. INTRODUCTION

The finite element technique is a relatively new method for stress analysis of structural continua.

Due to its almost limitless abilities this technique has recently been extensively used in structural analysis and considerable research has been conducted toward the development of elements and techniques satisfying the requirements of advanced structural technology.

The present work concerns the development of a general purpose digital computer program for three dimensional stress analysis, named TRISOP, and employing the finite element method.

The program developed, in its present form, is applicable to the solution of problems concerning linearly elastic structures, composed of homogeneous and isotropic materials, and subjected to static loading at constant temperature. The deformed states are restricted to small displacements and displacement gradients. The use of an equation solver of, in principle, infinite capacity devised by Cantin [Ref. 2] in conjunction with direct access storage devices enabled the memory space requirements to be independent of the size of the problem, the latter being limited only by the execution time and availability of storage devices.

At the time of initiation of the development of TRISOP the most promising finite elements, in the three dimensional

analysis field, were the family of isoparametric elements. Two such elements, namely the 20 (Quadratic) and 32 (Cubic) nodal points elements, are used in the program.

The present form of TRISOP is an improved version of a program coded by Professor Gilles Cantin of the Naval Post-graduate School and employing quadratic isoparametric finite elements. The major improvements brought to this initial form of TRISOP by the present version are, the extension to cubic elements, inclusion of predefined displacements in the boundary conditions, improved execution time and independence from the size of the problem.

The structure of the program follows a modular pattern in order to provide for further improvements and, possibly, adjustment to new developments in the field.

II. NATURE AND FORMULATION OF THE PROBLEM

The problem considered is a three dimensional stress analysis of a statically loaded structure under constant temperature. The analysis is restricted to deformed states within the elastic region and involving small displacement gradients. Consequently the linear strain tensor only need be considered and the strain-displacement relations can be expressed as:

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \qquad (2.1)$$

or in more compact form:

$$\{\varepsilon\} = [D]\{u\} \qquad (2.2)$$

Where both $\{\varepsilon\}$ and $\{u\}$ are point functions.

As a consequence of the above the principle of superposition applies to strains and displacements. The stress-strain law is assumed to be linear and thus the principle of superposition applies for stresses as well. Therefore,

displacements, strains and stresses of two distinct deformed states are algebraically additive resulting into a new compatible deformed state. Consequently initial stresses and strains need not be considered in the formulation of the problem. If such strains and stresses exist they will have to be superposed to the results of the solution which will be relative to a reference state, namely the state of the structure before the application of the load considered in the problem.

In addition to small displacement gradients, the analysis is restricted to displacements small enough that equilibrium and compatibility relations can be referred to the undeformed geometry of the structure (here and in what follows by undeformed state is meant the reference state).

Due to the restriction of small displacements the question of stability does not enter the analysis [Ref. 7]. The structure may be composed of more than one materials which are assumed to be homogeneous and isotropic. The constitutive laws for this problem can be expressed in matrix form as:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{Bmatrix} = \frac{E^*}{(1+\mu)(1-2\mu)} \begin{bmatrix} 1-\mu & \mu & & & & \\ \mu & 1-\mu & & & & \\ \mu & \mu & & & & \\ & & \frac{1-2\mu}{2} & & & \\ & & & \frac{1-2\mu}{2} & & \\ & & & & \frac{1-2\mu}{2} & \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{Bmatrix}$$

or in more compact form:

$$\{\sigma\} = [E]\{\varepsilon\} \qquad (2.3)$$

Where $\{\sigma\}$ is a point function, E* is Young's modulus and μ is Poisson's ratio.

A. FINITE ELEMENTS

The main feature of the finite element technique is the approximation of the elastic continuum possessing an infinite number of degrees of freedom by an assemblage of substructures, the finite elements, each possessing a finite number of degrees of freedom.

Among the various types, two elements of the isoparametric family were chosen for this development; namely, the 20 and 32 nodal point three dimensional elements. The concepts underlying the construction and use of these elements as well as the resulting properties will not be presented here except for those necessary for the presentation of the structure of the computer program. An extensive treatment of this family of finite elements as well as the isoparametric concept may be found in Refs. 4 and 8.

Two systems of reference are used throughout. First, a "local" non dimensional and, in general, non cartesian system (ξ,η,ζ) is used for convenience of computation of the element stiffness matrices. Then everything is transferred in a "global" cartesian system of reference (x,y,z) for the final computations of the assembled system.

A numbering convention for the element nodes is employed and is shown in Figures 1 and 2. A different convention is used for numbering the nodes of the assembled structure and is described in Section IV.A.

By use of shape functions $N_i(\xi,\eta,\zeta)$, describing the variation of element properties in the local system, every element property $\phi(\xi,\eta,\zeta)$ can be expressed in terms of its nodal values as:

$$\phi(\xi,\eta,\zeta) = \langle N_i \rangle \{\phi_i\}$$

Thus the displacements of points within an element will be given by:

$$\begin{Bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \cdots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \cdots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \cdots \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \end{Bmatrix}$$

or in more compact form:

$$\{u\} = [N]\{u_i\}$$

An extensive treatment of the shape function concept may be found in Ref. 8.

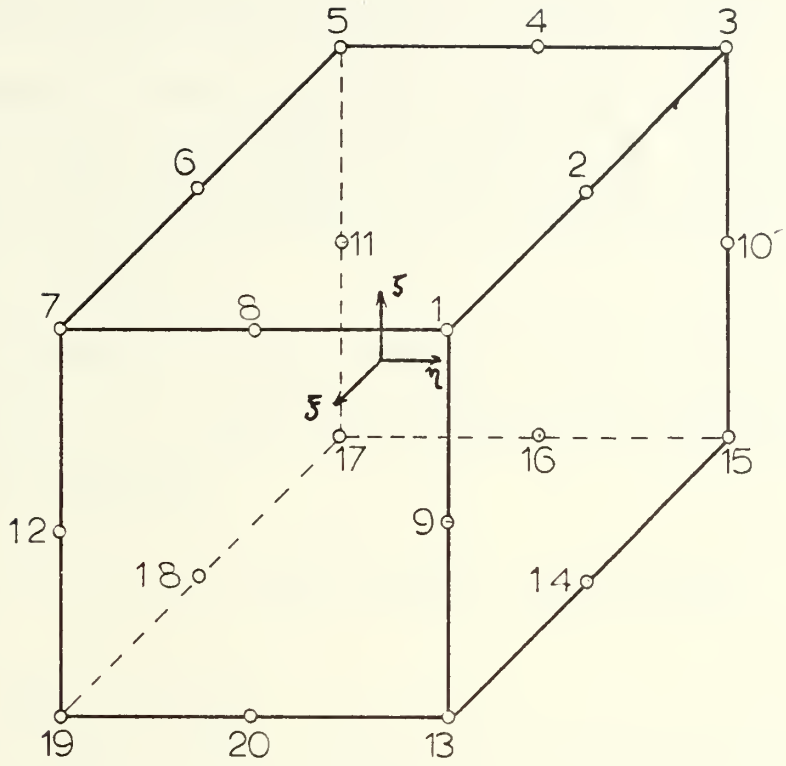


Figure 1.

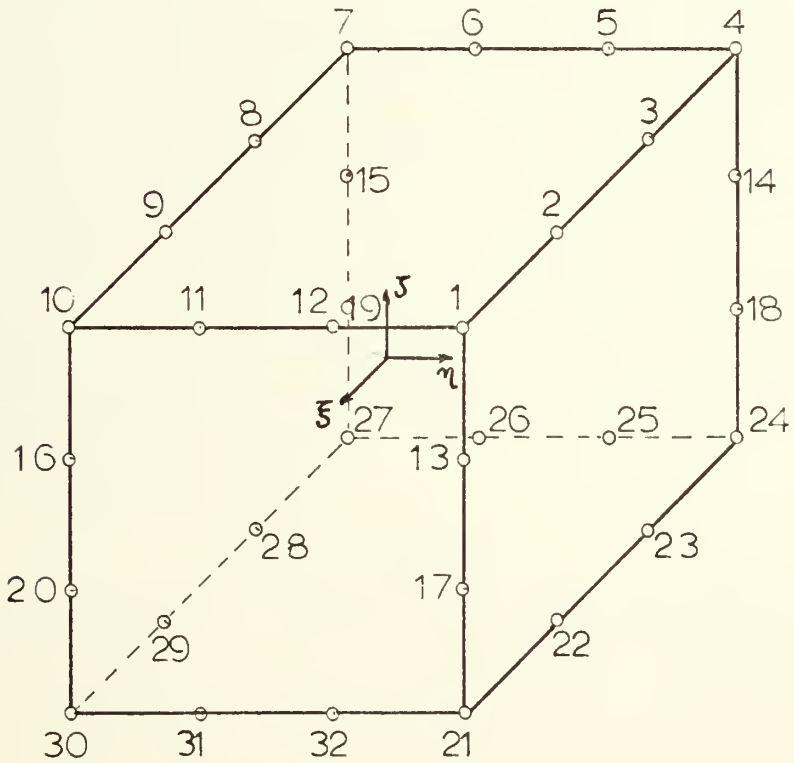


Figure 2.

The shape functions used in this development belong to the Serendipity family and are shown in Refs. 4 and 8.

B. ELEMENT AND TOTAL STIFFNESS MATRICES - CONSISTENT LOADS

Every individual finite element can be visualized as an independent structure and thus the derivation of a stiffness matrix is possible.

The derivation of the element stiffness matrix $[K_e]$ is shown in Ref. 1 where the following integral expression is finally obtained:

$$[K_e] = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [N^*]^T[E][N^*]\,\det[J]\,d\xi d\eta d\zeta \qquad (2.4)$$

where: $[N^*] = [D][N]$

and $[J]$ is the Jacobian matrix of the transformation.

This expression for the element stiffness matrix is ideally suited for Gaussian numerical integration. Computational considerations on the performance of this integration as well as the choice of integration points are presented in Ref. 1.

After formation of the stiffness matrix for each element, the total stiffness matrix [K] for the structure can be formed by superposition of corresponding stiffness coefficients.

The force-displacement relations for the whole structure can then be expressed as:

$$\{F\} = [K]\{u\} \qquad (2.5)$$

where $\{u\}$ is a vector containing all nodal displacements of the structure and $\{F\}$ a vector of consistent concentrated loads applied at the nodes and equivalent to the loading applied to

the structure. The determination of the consistent loads is not incorporated in the present solution and it is assumed that it is known.

Relations (2.5) represent a system of linear equations. When this system is initially formed, both {F} and {u} will in general contain known and unknown elements. By further manipulation the system can be brought to the standard form with all unknowns on one side and its solution will result into the determination of all unknown nodal displacements and forces (see Section III.B.5).

C. STRESSES AND STRAINS

By use of the strain-displacement relations (2.2) and the displacements obtained from the solution of system (2.5) the vectors of nodal strains can be determined.

However the application of relations (2.2) is only possible at the element level in the finite element theory. Thus the nodal strains for each element separately are first evaluated.

Further by applying the stress-strain relations (2.3) the element nodal strains are evaluated.

Except for nodes lying on the interfaces between elements of different materials, a stress analysis requires the determination of average nodal strains and stresses of all elements sharing common nodes. This is accomplished by simply averaging the algebraic values of the stress and strain components at each node.

III. DESCRIPTION OF THE COMPUTER PROGRAM

Computerized solutions to structural problems based on the finite element technique have, in the recent years, led to a considerable amount of research in three disciplines: structural analysis, numerical analysis and computer programming. Thus new elements and computational techniques are continuously being developed paralleled by the development of computing equipment. For this reason it is not unusual for such computer programs to become obsolete within a few years of their completion.

Under these circumstances, the development of a successful program is required to follow rules which will preserve its effectiveness as long as possible by an adjustment process. These rules are summarized to:

a. Standard coding and machine independence enabling the program to be used by various installations regardless of the computing equipment available, and

b. Modular construction allowing adjustment to new developments without requiring the construction of the entire program.

These principles have governed the development of TRISOP at all stages that brought it to its present form.

The structure of TRISOP follows closely the analytic development of Section II. (Also see Flow Diagram.)

The program is coded in standard FORTRAN and is machine independent except for the subroutine DISK via which transfer of data to and from direct access storage devices is accomplished. All calculations are performed in double precision in order to provide adequate accuracy in spite of the round off errors.

The memory space requirements are independent of the size of the problem, the latter being limited only by the size of direct access storage space available and execution time.

The extension of the capabilities of the program, as for example inclusion of thermal stress analysis and nonhomogeneous materials will easily be accomplished without interfering with parts of the program that are not altered by these changes.

A number of checking points was incorporated in order to assure the proper development of the execution. Whenever required, error and/or warning messages are given in the output. If the continuation of the solution is useless due to the error detected the execution for this problem is stopped and the solution of the next problem (if such exists) begins.

The most important variables used throughout the program are shown in Appendix A and a brief description of their function is given.

The basic characteristics of the two programs, for 20 and 32 nodal point elements, are the same. The minor existing differences are discussed in Appendix G where listings of both programs are given.

In what follows the function of the different modules of the program is explained with reference to the analytic development of the solution as it is presented in Section II.

A. CONTROL PROGRAM

The control program serves no other purpose but to initiate the calling sequence of the different modules of the program and read, from input data cards, the characteristics (number and length of records) of the data files as these are requested by the JCL[1] cards. These characteristics are later on (in subroutine INPUT) checked for correctness and adequacy.

B. SUBROUTINE MAINE

Subroutine MAINE is the actual main program. It is coded in subroutine form in order to provide for the solution of more than one problem in one run by simply restarting the calling sequence of the different modules. Its function consists of a step by step calling procedure. The only numerical operation executed by subroutine MAINE itself is the computation of the average nodal stresses and strains after the element nodal stresses and strains have been computed by subroutine STRESS. In the coding of subroutine MAINE a timing procedure has been incorporated giving at the output the execution time for each step. This is an optional feature not connected with the execution of the problem itself and can be removed if required. However, it provides useful data for studying the factors affecting the execution time of a problem.

---

[1] JCL = Job Control Language

The calling procedure and the function of the subroutines called are as follows:

1. Reading of Input Data

By calling subroutine INPUT all input data cards are read. First, the general characteristics of the problem (as number of elements, number of nodal points, etc.) are read and used for the computation of other characteristics (as total number of equations, number of blocks per column). All general characteristics of the problem are stored in a common storage area and used throughout the program by most of the subroutines. Subsequently the arrays of connectivity, nodal point coordinates, material properties, concentrated load data, and boundary conditions are read. All arrays except for the material properties (which is of pre-specified size) are stored in data files on direct access storage devices. This feature enables the program to be independent of the size of the problem. The bandwidth and the number of blocks per row of the total stiffness matrix are computed from connectivity data. All input data as well as those computed are printed out for checking purposes.

The nature of the program allows the solution of a great variety of problems. For this reason and in order to avoid limitations on the form of the input data most reading formats are not specified but left to the choice of the user by use of object time formats.

Within subroutine INPUT several checks are made in order to insure the consistency of some data. All such checks

throughout the program are accomplished by calling the proper entry of subroutine ERROR. The checks made by subrouting INPUT are the following.

a. Stop Trap for the Last Problem

The last problem of each run is followed by an indicator card, as explained in Section IV. Once this indicator card is read, the execution is stopped.

b. Number of Different Materials

The maximum permissible number of different materials present in the structure to be analyzed is ten. If a problem involves a number of materials greater than ten, the execution is stopped and an error message is given in the output.

c. Characteristics of Data Files

The characteristics of data files, namely the number and size of records for each file, are given to the program once for every run in the main program but their consistency is checked for every problem of the run. If the size of one or more of these characteristics is smaller than that required for the problem the execution is stopped for this problem and execution of the next problem (if such exists) begins. In case the size of one of the data files characteristics exceeds that required for the problem, a warning message is given at the output and execution continues.

2. Formation of Element Nodal Coordinate Arrays

The coordinates of nodal points are provided to the program via subroutine INPUT in a sequence following the node

numbering convention for the whole structure. However, at various points of the program the use of arrays containing the nodal coordinates of each individual element is required. The formation of such arrays is accomplished by subroutine SORT. The sequence of elements of these arrays follows the numbering convention of the element nodes. Each array formed is stored in a direct access data file.

3. Formation of Element Stiffness Matrices

The individual element stiffness matrices [Ke] are formed by subroutine FSTF in the following way.

The element stiffness matrix is given by the expression

$$[Ke] = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [N^*]^T[E][N^*]\, \det[J]\, d\xi d\eta d\zeta \quad (3.1)$$

First the matrix of elastic constants [E] is computed by calling subroutine ELPROP. In case the whole structure is made of one material this operation is executed only once since all elements have a common elastic matrix. In the case of more than one material matrix [E] is computed every time the element considered is of different material than the previous one. Subsequently subroutine CUB is called to perform Gaussian numerical integration. For each Gauss point subroutine FORMK is called by CUB and the integrant of expression (3.1) is formed in a step by step procedure and evaluated at the integration point. Once an element stiffness matrix is formed it is stored on disk.

4. Formation of Total Stiffness Matrix

The formation of the total stiffness matrix is performed in subroutine MERGE by superposition of the corresponding stiffness coefficients. In order to avoid the presence of the whole stiffness matrix in core, it is partitioned in blocks of pre-specified size. These blocks are formed one at a time by selecting the proper elements from the element stiffness matrices. Each element stiffness matrix is scanned in order to detect elements belonging to the block under formation. Every completed block of the total stiffness matrix is in turn stored on disk. The method of partitioning and storing of the total stiffness matrix is established in Ref. 2. A thorough study of this paper is essential in order to comprehend the procedure outlined here as well as to become familiar with the program itself.

5. Formation of Load Vector and Application of Boundary Conditions

After the formation of the total stiffness matrix the completion of system 2.5 requires the formation of vectors {F} and {u} and some further manipulation of the system in order to bring it in the standard form.

The load vector {F} is partitioned into a number of smaller vectors the size of which corresponds to the size of total stiffness matrix blocks. Each such vector is formed in sequence by use of the consistent load data, stored on disk from the first step of the execution (subrouting INPUT), and subsequently is stored on disk. Load vector elements

corresponding to unknown forces, namely reactions at the constraints, are replaced at this stage by zeros.

The boundary conditions imposed to the problem result in the presence of a number of unknown reactions, designated hereby F_i^*, and predefined displacements, designated by u_i^*. If such a boundary condition is applied to the ith element of the force and displacement vectors the system will have the form:

$$\begin{Bmatrix} F_1 \\ \vdots \\ F_i^* \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1i} & \cdots & K_{1n} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \cdots & K_{ii} & \cdots & K_{in} \\ \vdots & & \vdots & & \vdots \\ K_{n1} & \cdots & K_{ni} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_i^* \\ \vdots \\ u_n \end{Bmatrix}$$

This system is equivalent to a good degree of approximation to the system

$$\begin{Bmatrix} F_1 - K_{1i}u_i^* \\ \vdots \\ K_{ii}u_i^* \\ \vdots \\ F_n - K_{ni}u_i^* \end{Bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1i} & \cdots & K_{1n} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \cdots & B & \cdots & K_{in} \\ \vdots & & \vdots & & \vdots \\ K_{n1} & \cdots & K_{ni} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ F_i^{**} \\ \vdots \\ u_n \end{Bmatrix}$$

Where B is a number of several orders of magnitude greater than the greater stiffness coefficient and $F_i^{**} = -\frac{F_i^*}{B}$

The products $F_i^{**}K_{ii},\ldots,F_i^{**}K_{ni}$ are negligible in size compared to the other coefficients. After the system is solved the unknown reaction F_i^* is determined by: $F_i^* = -BF_i^{**}$

This is a well known method of solution of mixed systems and it is felt that no further explanation is required [Ref. 5].

The process of application of the boundary conditions and transformation of the system to a form suitable for solution is performed by subroutine BCOND. In the case where all predefined displacements are equal to zero the process can be considerably simplified. In this case the final form of the system is:

$$\begin{Bmatrix} F_i \\ \vdots \\ 0 \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1i} & \cdots & K_{1n} \\ \vdots & & \vdots & & \vdots \\ K_{i1} & \cdots & B & \cdots & K_{in} \\ \vdots & & \vdots & & \vdots \\ K_{n1} & & K_{ni} & & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ F_i ** \\ \vdots \\ u_n \end{Bmatrix}$$

It is apparent that the only operation required is the substitution of the diagonal element K_{ii} by B.

For this reason subroutine BCOND was provided with two entries, one for the case of all zero predefined displacements and one (BCOND2) for the case of one or more nonzero predefined displacements. A number $B=10^{20}$ is used and is considered sufficient for the purpose of the program. The operations are carried out in a step by step procedure in a sense that at maximum two stiffness matrix coefficient blocks and their corresponding load vectors are simultaneously present in core.

6. Solution of the System - Determination of Reactions

The solution of the system is performed by subroutine SOLVE. The technique employed was devised by Professor Gilles Cantin of the Naval Postgraduate School and presented in Ref. 4.

In the process of solving the system of equations the inversion of diagonal blocks of the stiffness matrix is required. This inversion is performed by subroutine SYMINV. Before inversion of the block is performed it is checked whether it is singular or nearly singular. The trace of the matrix is evaluated (excluding diagonal elements containing the number $B=10^{20}$) and all diagonal elements are compared to the trace multiplied by a factor of 10^{-12}. If a diagonal element is found smaller than this product the block is characterized as nearly singular. The execution continues but a proper indication is given by SYMINV to SOLVE which in turn gives a warning message at the output via subroutine ERROR. The choice of the factor 10^{-12} was based on the number of significant digits provided by an IBM/360 machine on which the program was tested. Should the program be used on a different machine, alteration of this number should be considered in a sense that the difference between the exponent of the factor and the humber of significant digits provided by the machine should be adequate to provide for loss of accuracy due to round off errors. An extensive presentation of the concepts underlying this idea is beyond the scope of this work.

If one diagonal element is found to be equal to zero the block is singular and further continuation of the execution is impossible. Again a proper indication is given by SYMINV to SOLVE which in turn gives an error message at the output via subroutine ERROR and the execution is stopped for this problem. Consequently the calling procedure is restarted from the beginning and the execution of the next problem begins (if such exists).

Upon inversion of each diagonal block [A] of the stiffness matrix a condition number C is evaluated for this block within SYMINV. This condition number is defined as $C = \|A\| \cdot \|A^{-1}\|$ whereby $\|A\|$ is meant a norm of [A]. This condition number is printed out, and provides an upper bound on the expected relative error of the results of the problem. The difference between the number of significant digits provided by the machine used and the maximum exponent of the condition number gives approximately the number of significant digits in the displacement results. Further information on the significance of the condition number can be found in Ref. 6.

Following the solution of the system of equations the reactions are evaluated by subroutine DISP as explained in Section B5 but with an opposite sign. Consequently the nodal displacements and reactions are printed out by DISP.

7. Evaluation of Nodal Strains and Stresses

The nodal displacements as found by the solution of the system 2.5 are stored in vector form and in a sequence following the numbering convention of the nodes for the whole structure (Section II.A).

The determination of element nodal displacements by use of relation 2.2 requires a displacement vector containing the displacement components of all nodes of the corresponding element in a sequence following the numbering convention for individual elements (Section II.A). The formation of such vectors is accomplished by properly sorting the displacement components via subroutine DISK (entry WDISKB) called by DISP.

Subroutine STRESS evaluates the element nodal strains and stresses by use of relations 2.2 and 2.3. The evaluation of the product [D]{u} requires the use of subroutine FORMK. The computational details of this operation are explained in Ref. 1.

Printing of element strains and stresses is optional since they provide useful information only for nodes on interfaces between elements of different materials.

Evaluation and printing of average nodal strains and stresses is performed by subroutine MAINE and is the concluding operation in the solution of each problem.

C. SUBROUTINE DISK

Transfer of data to and from the data files stored on direct access storage devices is accomplished by subroutine DISK. This subroutine uses multiple entries, each entry performing a different operation. Calling of DISK has been reduced to a minimum to avoid useless expense of execution time in data transfer. The most critical entries are those used for transferring element stiffness matrices and total stiffness matrix

blocks. The reason being the size of the arrays transferred and the number of transfers required. Reduction of the transfer time requires the use of the maximum possible record length. The size of element stiffness matrices is known for each type of element, thus the optimum record length is determined and the corresponding entries are coded on this basis. However, the block size in the equation solver is determined by the user and thus the optimum record length must be determined by the user (see Section IV). The function of the various entries is explained below. Entry names starting with the letter W denote transfer of data to a data file and names starting with the letter R denote transfer of data from data files to core.

RDDISK, WRDISK: File number 7. Transfer of total stiffness matrix blocks.

RDISK1, WDISK1: File number 8. Transfer of element stiffness matrices.

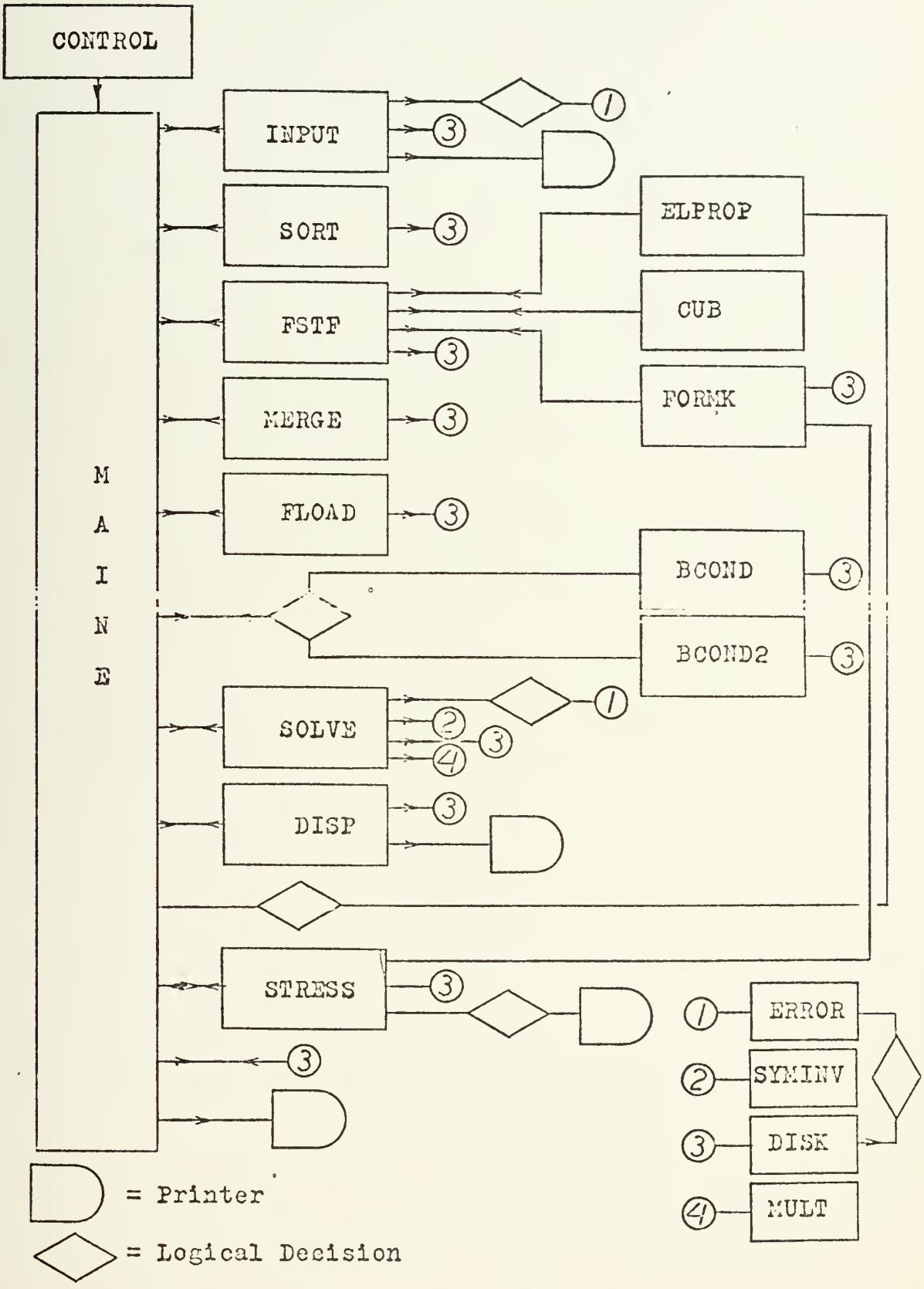
RDISKC, WDISKC: File number 10. Transfer of nodal coordinate component vectors COORD (3)

RDSKR1, WDSKR1: File number 20. Transfer of load vector blocks when stored in work vector RB1. Similarly for RDISKR2, RDISKR3, WDISKR2, WDISKR3.

RDISKB, WDISKB: File number 21. Transfer of individual node displacement components vectors. The random access data file thus formed is used for the formation of element nodal displacement vectors.

RDISKS, WDISKS: File number 12. Transfer of vector SSSS1 (See Appendix A.)

FUNCTIONAL FLOW DIAGRAM



IV. PROBLEM AND INPUT DATA PREPARATION

In this section the various steps in modeling a problem to be solved and the preparation of input data for the computer program are described.

A. STRUCTURE MODELING

The first step in applying a finite element solution to a problem is the selection of an appropriate mesh. In many cases an extrapolation of the results will be required and hence more than one meshes will have to be selected.

The mesh size selection does not follow any predetermined rules and will, in general, depend on the nature of the problem and the judgement of the analyst.

An arbitrary numbering convention must be adopted for the nodal points of the entire structure (in distinction with the numbering convention for the element nodes shown in Section II). The relation of node numbers in the two numbering systems establishes the connectivity matrix described in Section IV.B.1.

Though no restriction is imposed on the selection of the numbering convention for the entire structure, it has been shown in practice that an intelligent selection may considerably reduce the bandwidth of the total stiffness matrix.

B. INPUT DATA PREPARATION

Another factor to be determined by the analyst is the size of the blocks of coefficients in the equation solver. The block size affects the execution time required for solution of the system of equations by subroutine SOLVE. Some experimentation has been conducted on this subject by Cantin and is presented in Ref. 2. On the basis of this experimentation it seems that the larger the block size the lesser the execution time is. However, the subject has not, as yet, been thoroughly investigated and no firm rules can be stated at present. TRISOP allows the use of any block size. However, if a block size smaller than the size of the element stiffness matrix is selected, the dimensions of the various arrays of common block B3 may have to be changed.

After selection of the block size the number of records per block NREC7 and record length of file number 7 can be determined. Maximum transfer efficiency requires that NREC7 is minimum and thus the record length the maximum possible for the machine used. Subsequently the remaining input data can be determined. In the following a description of the non self-explanatory data is given.

1. Connectivity Matrix

The size of this matrix is NELxNPEL. Each row corresponds to one element and represents the correspondence of node numbers in the two numbering systems, namely, the element node numbering system and the structure node numbering

system. The values of the elements of each row are the node numbers in the entire structure system and are sequenced according to the element node numbering system. Thus, the second element of the fifth row represents node number two of finite element number five and its value is the number of this node in the structure system.

2. Nodal Point Coordinates

The nodal point coordinates are defined in the global system of reference. Cartesian or cylindrical coordinates may be used. These coordinate systems are defined in Fig. 3. All three coordinates of each node must be defined in the

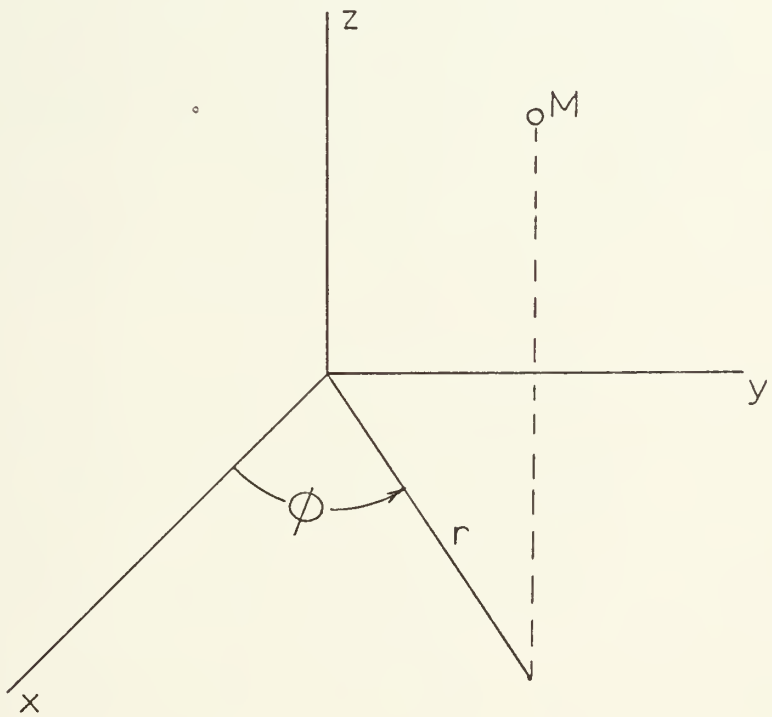


Figure 3.

same system. However, different nodes may be defined in different coordinate systems. Each set of nodal coordinates must be accompanied by a code number (KND) indicating whether cartesian or cylindrical coordinates are used.

3. Material Properties

The maximum permissible number of different materials is ten. In case the structure is composed of more than one materials a code number (NCON21) must be assigned to each material (NCON21=1,2,..., n, where n is the number of materials present in the structure). Material properties are given in the form of matrix ELDAT. Each row represents a different material and its elements are in sequence, Young's modulus, Poisson's ratio and coefficient of thermal expansion. The present form of TRISOP does not provide for thermal stress analysis and thus no value is required to be given to the coefficient of thermal expansion.

C. DATA CARDS

In order to preserve the flexibility of the input no reading formats are specified in the program except for titles, object time format arrays and general characteristics of the problem. Object time formats, stored in the array FMT, are used for reading input data. The same formats are used for echo-printing of the input data. In the description of input data cards the variable FMT denotes the object time format under which the corresponding data wll be read. One data card is required per format. Formats may be arbitrarily

specified by the user. However it is recommended that only one data card be used for the variables described in Sections IV.C.5.c; 6.c; 7.c; 8.c; 9.c and 10.c. The number of cards required per group of data is based on the above recommendation.

Every group of data cards is preceded by a card containing alphanumeric information. Exception is made only for the connectivity matrix of cubic elements where two such cards are required. The contents of these cards are stored in the array TITLE and appear as identification headings at the output.

Both TITLE and FMT are read under a (10A8) format.

1. Characteristics of Data Files

These data are given only once for each run and are read under a (7I10) format. Six cards are required on which the values of record length and number of records for the various data files are given in the sequence: NR7, LRW7, NREC7, NR8, LRB8, LRW8, NR9, LRB9, NR10, LRB10, LRW10, NR12, LRB12, LRW12, NR13, LRB13, LRW13, NR14, LRB14, NR15, LRB15, NR16, LRB16, NR17, LRB17, NR19, LRB19, NR20, LRB20, LRW20, NR21, LRB21, LRW21, NR22, LRB22.

These cards can be produced by use of the auxiliary program JCL as described in Appendix B. The required values of the above variables are given in the same Appendix.

2. Initiation Card

Format: 1A8

Number of cards per problem: one

The word START must be punched starting at column one.

3. Problem Identification

Format: (10A8)

Number of cards per problem: one

Any alphameric information identifying the problem and punched in columns 2-80.

4. General Characteristics of the Problem

Format: (10I5)

Number of cards per problem: one

Columns	Variable	Meaning
1-5	NEL	Total number of elements
6-10	NDPT	Total number of nodal points
11-15	NS	Stiffness matrix block size
16-20	NPEL	Number of nodes per element
21-25	NMAT	Number of different materials
26-30	NCLD	Total number of nodes with concentrated loads
31-35	NPBC	Total number of nodes with boundary conditions
36-40	NPBC2	Number of nodes with non zero predefined displacements
41-45	NLINE	Number of lines to be printed per page for strain and stress tables
46-50	KEL	KEL=0: No printing of element nodal strains and stresses is required. KEL≠0: Printing of above is is required.

5. Connectivity Matrix

a. TITLE

b. FMT

c. Connectivity data.

Number of cards: NEL

Variables: NCON= Array of connectivities for one element (one row of connectivity matrix)

NCON21= Material code number for the corresponding element (if required)

The connectivity matrix is read row-wise one row at a time. In the case of structures composed of more than one material, the last element of NCON must be followed by NCON21.

6. Nodal Coordinates

a. TITLE

b. FMT

c. Coordinate data

Number of cards: NDPT

Variables: IEL= Node number in the entire structure numbering system.

COORD= Array of coordinates of node IEL in the global system. Coordinates are given in the sequence x, y, z or r, ϕ, z.

KND= Code number indicating cartesian or cylindrical coordinates.

KND=0: Cartesian coordinates

KND$\neq$0: Cylindrical coordinates.

7. Material Properties

a. TITLE

b. FMT

c. Material property data

Number of cards: NMAT (maximum ten)

Variables: N= Material code number

ELDAT= Array of material properties.

8. Nodal Consistent Load Data

a. TITLE

b. FMT

c. Load data

Number of cards: NCL= Node number in the numbering system of the entire structure

CLOAD1= Array of concentrated load components in the global system. Load components are given in the sequence x,y,z.

Load data cards must be in ascending order of NCL values.

9. Boundary Condition Codes

a. TITLE

b. FMT

c. Codes

Number of cards: NPBC

Variable: NBC= Array of boundary condition codes for each node. The first element is the node number and the remaining three represent the boundary

conditions applied in the x,y,z direction respectively.

The codes employed are:

0: Free node.

1: Node constrained to zero displacement.

2: A predefined displacement exists.

The boundary condition code data cards must be sequenced in ascending order of node number.

10. Predefined Displacements (if required)

a. TITLE

b. FMT

c. Displacement data.

Number of cards: NPBC2

Variables: NDBC = Node number.

BCON = Array of predefined displacement components in the global system, given in the x,y,z sequence.

Predefined displacement data cards must be sequenced in ascending order of the values of NDBC.

In case no predefined displacements exist in a problem (NPBC2=0) all cards described in this paragraph must be omitted.

11. Subsequent Problems - Stop Trap for Last Problem

In case more than one problem is to be executed in one run, the data information described in paragraphs IV.C,2 to IV.C.10 must be provided for each problem. The initiation

card (START) of each problem must immediately follow the last card of the previous problem.

The last problem must be followed by the following cards which serve as a stop trap.

a. One card with the word "START" starting in column one.

b. One card containing any alphanumeric information (or blank).

c. One card with any negative integer punched in columns 1-5.

In order to provide for a better understanding of the above documentation sample inputs for two problems, one utilizing quadratic and one cubic elements are presented in Appendix C.

D. DIRECT ACCESS DATA FILES REQUIREMENTS

The method of specification for the direct access file requirements will vary from one installation to the other. The particular features of the different installations will affect the form of subroutine DISK and the JCL cards. In Appendix B these requirements are described for the IBM 360/67 operating system. The space requirements for each file as described in Appendix C are those to be used in establishing the form of subroutine DISK and JCL cards for any machine.

V. SOLUTION OF A SIMPLE PROBLEM

When the initial form of TRISOP, for quadratic elements, was coded by Professor Cantin, it was tested on simple problems, and results were obtained and compared with classical solutions.

Testing of the present development of TRISOP was based on these problems and no further experimentation was conducted.

In order to show the function of the program, as well as some observations affecting the use of it, the solution of a simple problem is presented in this section.

The problem selected is that usually presented in classical texts of strength of materials as a simply supported beam of uniform cross section subjected to a concentrated load in the middle of the span.

The model used for this problem for the quadratic element formulation is shown in Fig. 4. The concentrated load is substituted by a line load. Nodes 1, 2 and 3 (here arbitrarily numbered) are constrained to zero displacement in all directions and nodes 4, 5 and 6 are allowed to move in the z direction only. The model for cubic elements is the same with the difference that the constrained nodes are four at each end instead of three.

The consistent loads corresponding to the line load applied were found as required by the program. Such loads appear only at the nodes of the midsection of the beam and are shown in Fig. 6.

Five different meshes were employed in order to show the variation of results with mesh size. The meshes chosen were 2, 4, 6, 8 and 10 elements in the z direction for both quadratic and cubic elements.

Numerical integration was carried by use of 2, 3 and 4 Gauss points (in different runs) for quadratic elements and 3, 4 and 5 points for cubic elements. Listings of the subroutines used are shown in Appendix I.

Integration with 3 and 4 Gauss points for quadratic elements and 4 and 5 points for cubic elements gave essentially the same results for this problem. For problems using elements with curved boundaries the results will vary with the number of Gauss points used. The solution with quadratic elements using 2 integration points was numerically unstable and no results were obtained. Numerical results for the maximum displacement at the midsection are given in Tables I and II. The same results are shown graphically in Fig. 7 for both quadratic and cubic elements. In the same figure, the execution time for subroutine FSTF and SOLVE, which consume most of the execution time for each problem, is shown. Figure 8 shows graphically the displacements and execution time obtained from the solution using cubic elements.

A. ALTERNATE FORMULATION

In order to observe some further features of the finite element solution, the same problem was solved formulated in a different way.

By use of the existing symmetry in the z direction, only half of the beam was considered as shown in Fig. 5. All nodes lying in the plane of symmetry were constrained to a zero displacement in the x direction. The presence of additional constraints enabled the use of two integration points. Numerical values for the maximum displacement in the midsection are shown in Table III. The same results as well as execution time are shown in graphical form in Fig. 9.

B. DISCUSSION OF RESULTS

Observation of the results presented here reveals that mesh refinement leads to improved results which eventually will converge to a certain value. Use of less accurate numerical integration by reducing the number of integration points leads to improved results and considerable reduction of execution time. Also the use of existing symmetry and in general different formulations of the problem may prove to be beneficial.

Results obtained by use of cubic elements are considerably better than those of quadratic elements. However the execution time is appreciably higher for cubic elements and it is a matter of judgement whether cubic or quadratic elements should be used for each specific problem.

These are observations essentially made on the results of a single and basically simple problem. Thus no firm rules regarding the use of TRISOP can be established and further experimentation with different problems has to be conducted for this purpose.

TABLE I

Quadratic Elements - Maximum Displacements in x 10^{-4}

ELEM.	2	4	6	8	10
CUB 3 or CUB 4	5.761	6.847	6.992	7.026	7.038

TABLE II

Cubic Elements - Maximum Displacements in x 10^{-4}

ELEM.	2	4	6	8	10
CUB 5 or CUB 4	7.045	7.055	7.062	7.067	7.070
CUB 3	7.045	7.067	7.068	7.069	7.072

TABLE III

Quadratic Elements - Alternate Formulation

Max. Disp. in x 10^{-4}

ELEM.	2	4	6
CUB 4 or CUB 3	5.761	6.847	6.992
CUB 2	7.191	7.066	7.053

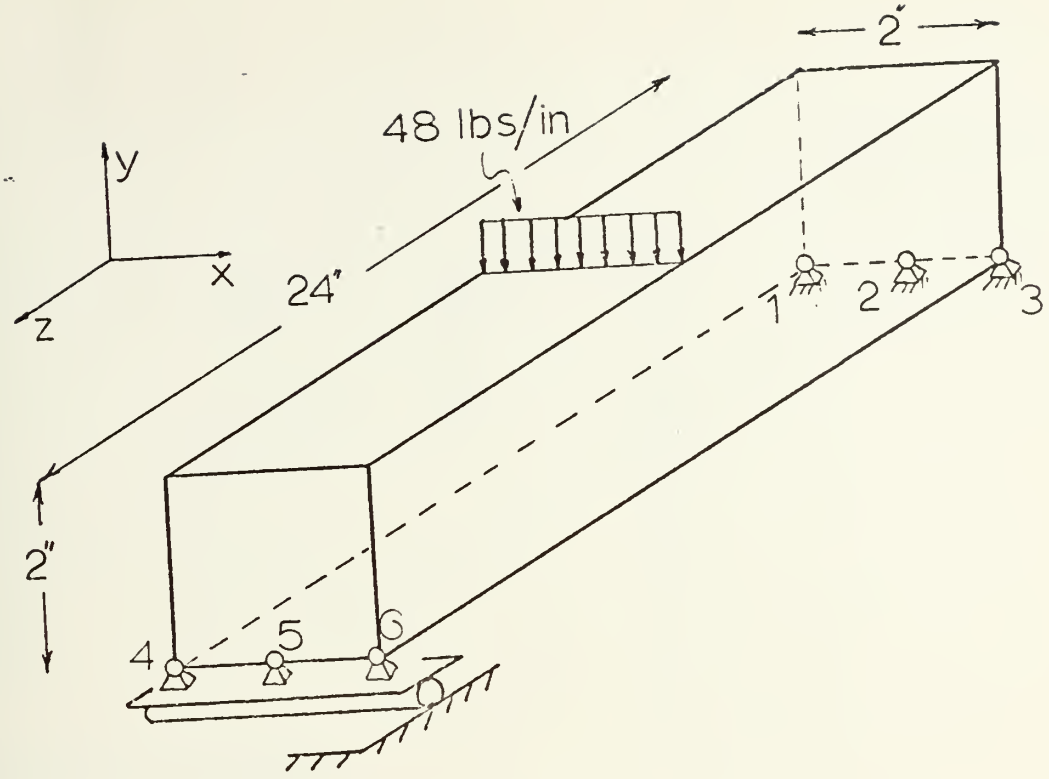


Figure 4

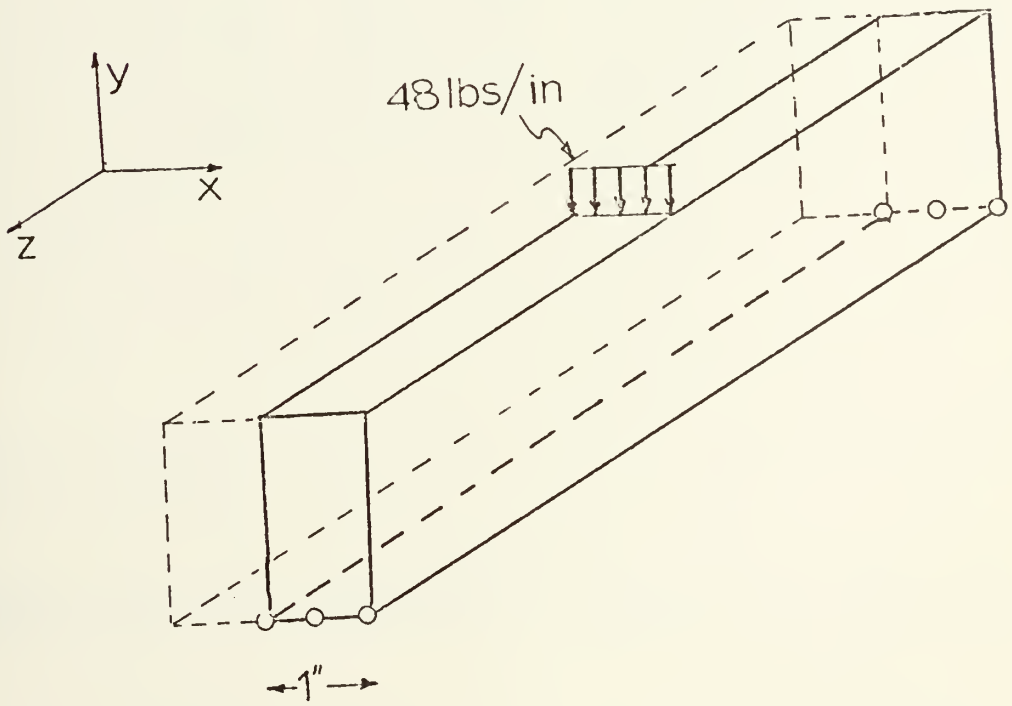
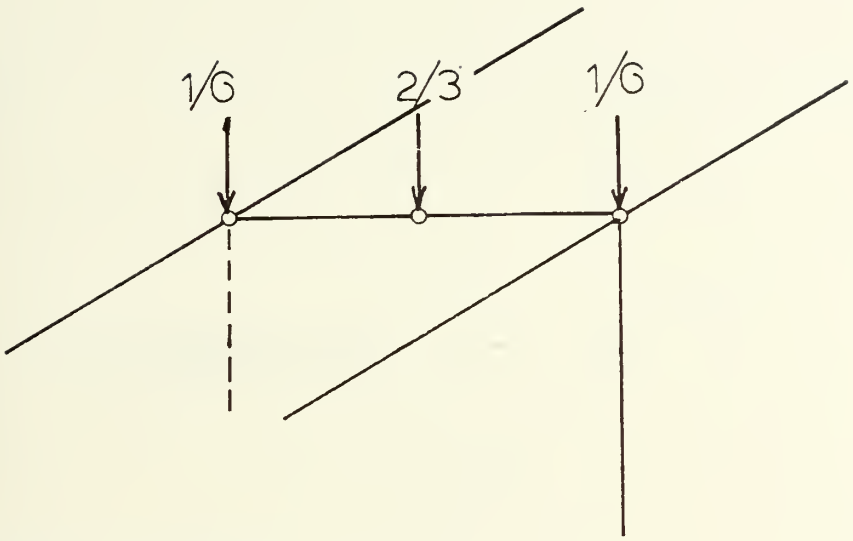
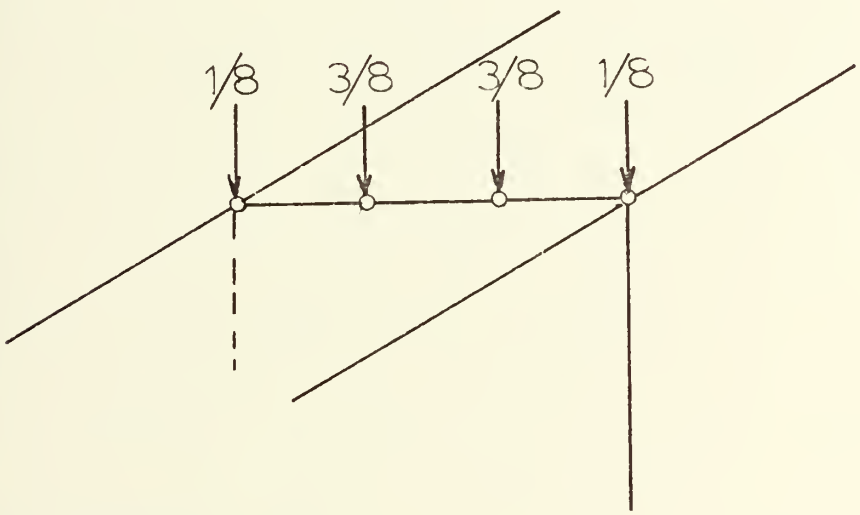


Figure 5



Quadratic elements



Cubic elements

Numbers represent proportions of total load applied on each node.

Figure 6

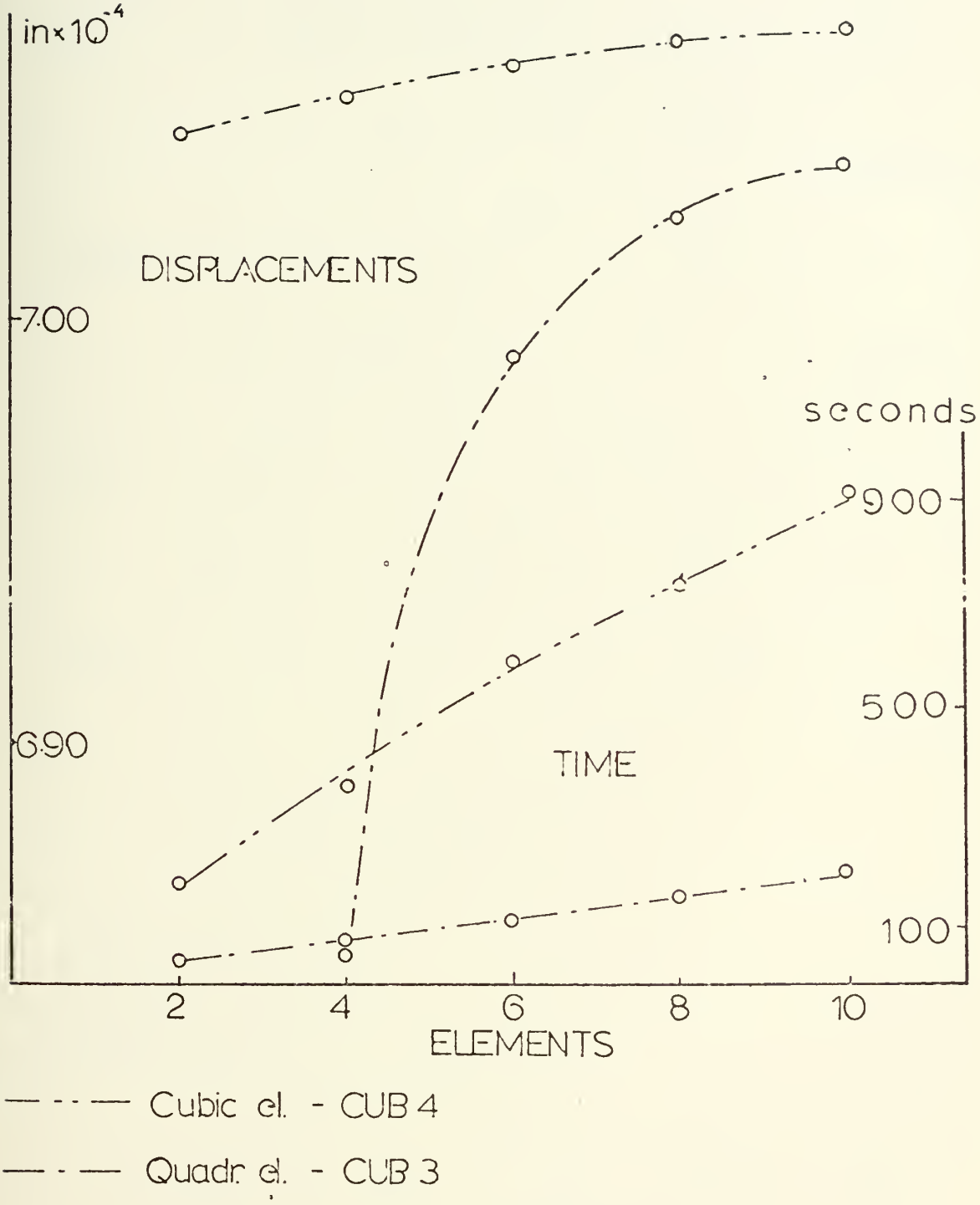

in×10⁻⁴
DISPLACEMENTS
7.00
6.90
seconds
900
500
100
TIME
2
4
6
8
10
ELEMENTS
Cubic el. - CUB 4
Quadr. el. - CUB 3

Figure 7

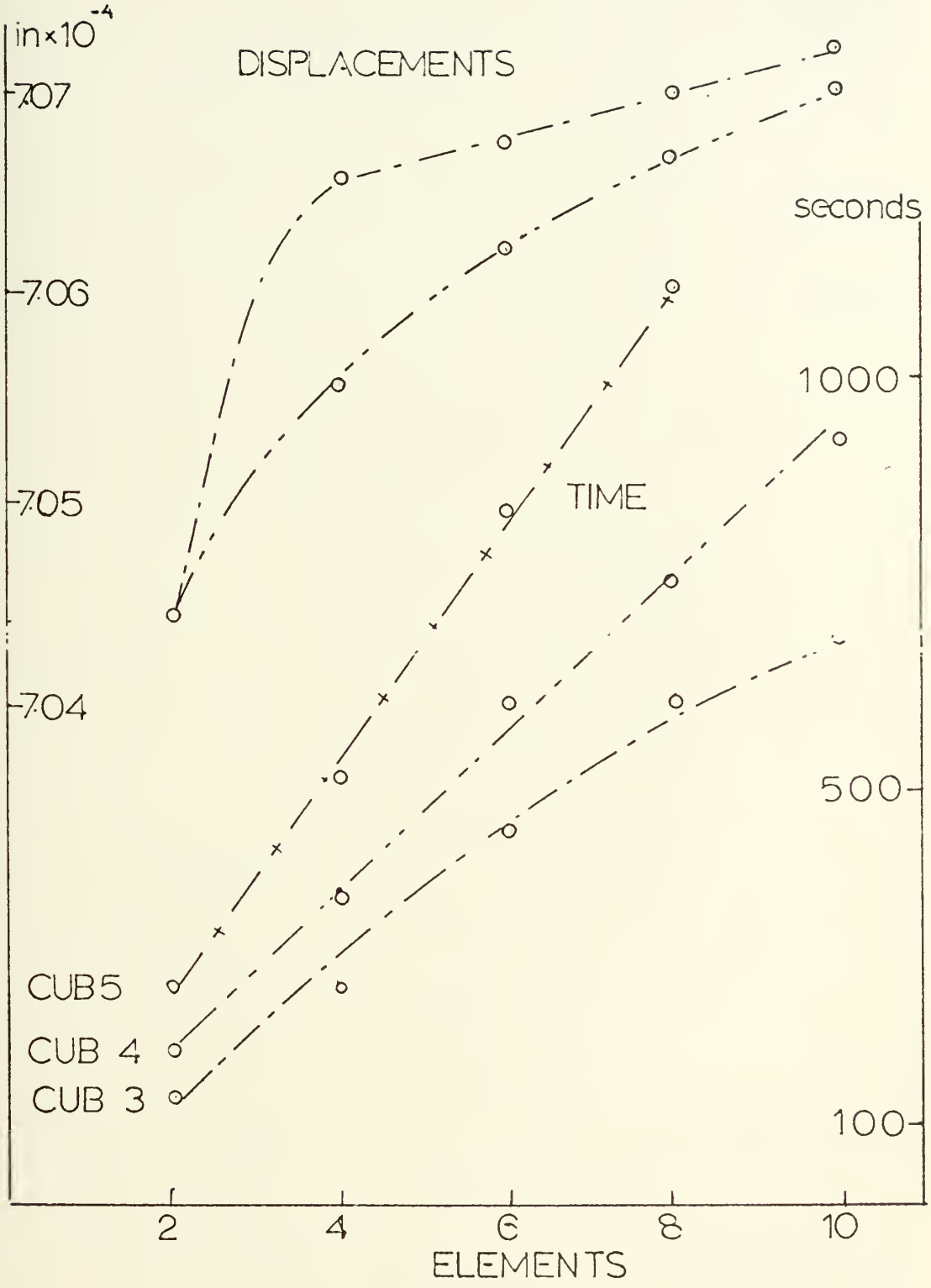

in×10⁻⁴
DISPLACEMENTS
7.07
7.06
7.05
7.04
seconds
1000
500
100
TIME
CUB 5
CUB 4
CUB 3
2
4
6
8
10
ELEMENTS

Figure 8

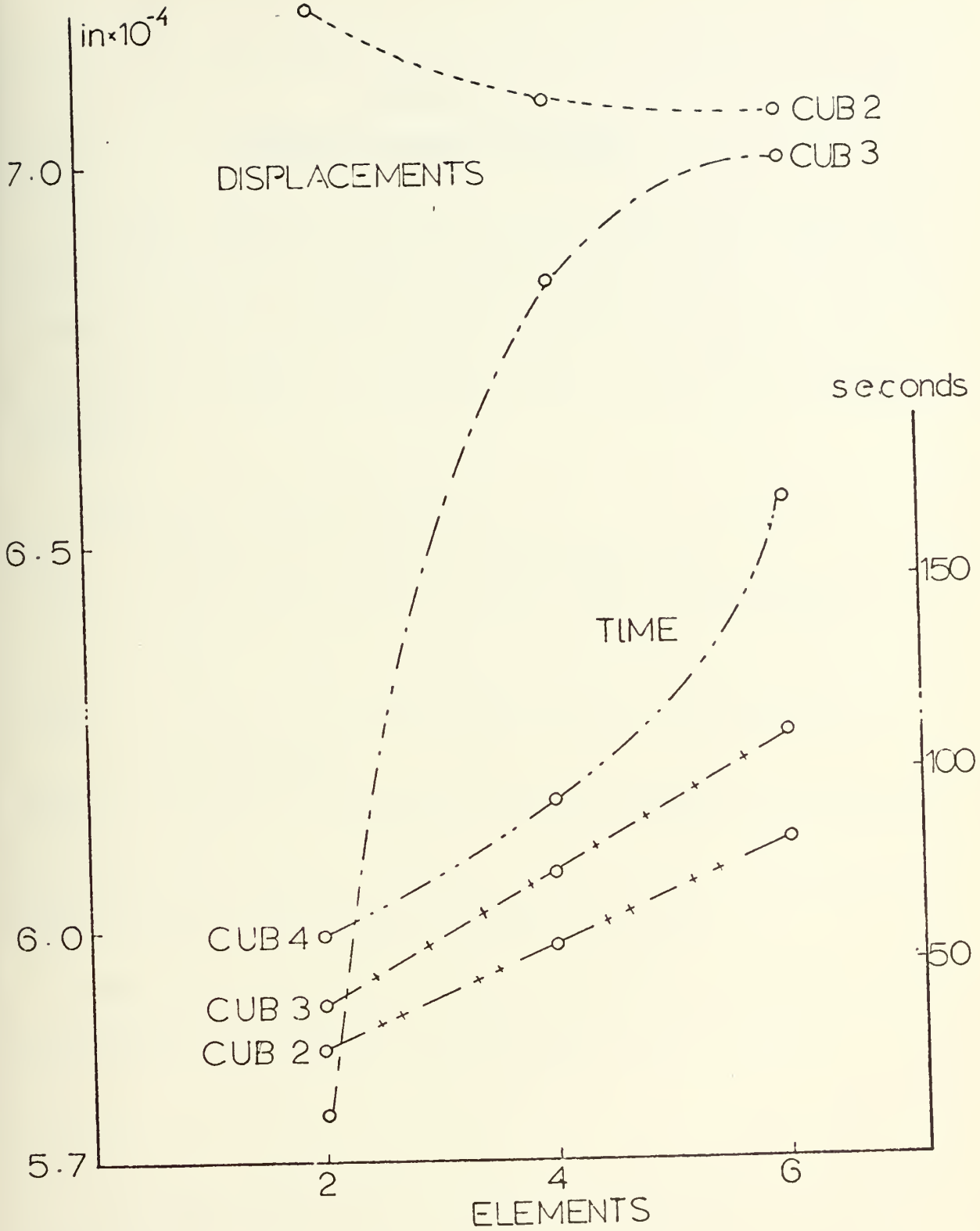

in×10⁻⁴
7.0
6.5
6.0
5.7
DISPLACEMENTS
CUB 2
CUB 3
CUB 4
CUB 3
CUB 2
TIME
seconds
150
100
50
2
4
6
ELEMENTS

Figure 9

APPENDIX A

COMPUTER PROGRAM VARIABLES

In this Appendix the most important variables appearing in the programs are listed together with a brief description of their function. The variables of each group are sequenced in alphabetic order. Arrays are shown with the dimensions required.

A. DATA FILES CHARACTERISTICS

NRX : Number of records of file number "X"

LRBX : Length of records of file number "X" measured in bytes

LRWX: : Length of records of file number "X" measured in single words.

NREC7 : Number of records per total stiffness matrix block

NRX, LRBX AND LRWX are used only for checking the adequacy of disk storage space requested by the JCL cards.

NREC7 specified by the user, is used in subroutine DISK and controls the transfer of stiffness matrix blocks.

B. VARIABLES APPEARING IN 'COMMON' STATEMENTS

AK1(NS,NS) : Work space used primarily for storage of stiffness matrix blocks when transferred in core. Alternatively is used for storing

equivalenced arrays and thus reduce the space requirements. Similarly for AK2 and AK3.

BCON(3) : Array of non zero predefined displacement components for one node. It is used only in entry BCOND2 of subroutine BCOND. The complete array of non zero predefined displacements is stored on disk and only the components for one node at a time are present in core.

CLOAD(4) : Concentrated load data for one node. The first three elements represent the components of the load and the fourth the node number. It is used only in subroutine FLOAD. The complete array of concentrated load data is stored on disk and only data for one node at a time are present in core.

COORD(3) : Array of global coordinates for one node. The complete array of nodal coordinates is stored on disk in node sequence following the numbering convention for the whole structure. Coordinates of only one node at a time are present in core.

COREL(NPEL,3) : Array containing the global coordinates for all nodes of one element. These arrays are formed by SUBROUTINE SORT one at a time and subsequently stored on disk. Only one such array is present in core every time.

ELAST(616) : Matrix of elastic constants of elements formed by subroutine ELPROP.

ELDAT(10,3) : Array containing information on properties of materials composing the structure. It is used in ELPROP for the formation of ELAST.

KEL : Index number indicating whether printing of element nodal strains and stresses is required or not.

MM : Number of stiffness matrix blocks per row (see Ref. 2)

NBC(4) : Boundary condition information for one node. The first element represents the node number and the remaining the boundary condition codes. The complete set of such arrays is stored on disk in node number sequence and only one is present in core at a time.

NBCH : Highest node number at which a boundary condition is applied.

NBCL : Lowest node number at which a boundary condition is applied.

NCLD : Total number of nodes at which a concentrated load is applied.

NCON(NPEL) : Array of connectivities for one element. This array shows the correspondence of element node numbers in the two numbering systems employed, namely, element numbering system and entire structure numbering system. The values of

the elements of the array represent the node numbers in the structure numbering system and are sequenced according to the element numbering system. Thus the third element of the array represents node number three of the element and its value represents the number of this node in the structure numbering system. The complete set of these arrays is stored on disk and only one at a time is brought into core.

NCON21 : Indicates the material of the corresponding element and is used in the case the structure is composed of more than one material. The values of this variable must conform with the material number given in array ELDAT.

ncount : Number of coefficients per stiffness matrix block.

NDF : Number of spacial degrees of freedom. In the present form of the program its value is constant and equal to three. However, it is used throughout the origin as a variable in order to provide for future developments of the programs as for example degeneration to two dimensional elements.

NDFP : Constant equal to: NDF+1

NDPT : Total number of nodal points in the structure.

NEL : Total number of finite elements used in the problem.

NEQ : Total number of equations of system (2.5).

NLINE : Number of lines to be printed per page in the output for displacement, reactions, strains and stresses.

NMAT : Number of materials of which the structure is composed.

NN : Number of stiffness matrix blocks per column (See Ref. 2).

NPBC : Total number of nodes at which boundary conditions are applied.

NPBC2 : Total number of nodes at which non zero predefined displacements are applied.

NPEL : Number of nodes per element.

NREC7 : Number of records per stiffness matrix block.

NS : Stiffness matrix block size.

NST : Size of element stiffness matrix.

NSTF : Number of coefficients per element stiffness matrix.

SSSS1(7) : Work vector used for transferring to and from data files the algebraic sums of stress or strain components contributed by elements sharing a common node. The seventh element of the array is used to count the number of occurrences of the node. Such arrays are formed in subroutine STRESS and further used in MAINE for the evaluation of average strains and stresses.

UJNT(3) : Array of displacement components for one node. The complete set of such arrays is stored on disk and

displacements for only one node at a time are present in core.

C. OTHER IMPORTANT VARIABLES

In order to avoid redundant definitions, the most important variables of each subroutine that have not been presented in Sections A and B of this Appendix are explained below.

1. Subroutine ELPROP

E : Young's modulus

PR : Poisson's ratio

2. Subroutine CUB

STK(NST,NST) : Element stiffness matrix

X,Y,Z : Gauss points coordinates

3. Subroutine FORMK

AJ(NDF,NDF) : Jacobian matrix

AK(NST,NST) : Integrant of stiffness matrix integral formula

B(6,NST) : Matrix [N*] = [D][N]

CORDG(NPEL,NDF): Element nodal coordinate array in the local system

DNX(NDF,NPEL) : Array of derivatives of shape functions with respect to x,y,z.

Wl(NDF,NPEL) : Array of derivatives of shape functions with respect to ξ,η,ζ.

4. Subroutine BCOND

NBC : Stiffness matrix row number corresponding to the 'z' component of the boundary condition examined.

IBL,IBH : Lowest and highest row number in the block examined.

INDEX : Number of boundary conditions applied.

AKW1(NCOUNT)
AK1(NS,NS) : Diagonal stiffness matrix block.

AKW2(NCOUNT)
AK2(NS,NS) : Offdiagonal stiffness matrix block.

RB1(NS) : Load vector corresponding to row 'I'.

RB2(NS) : Load vector corresponding to blocks symmetric to the offdiagonal blocks of row 'I'.

MM1 : Specifies the extreme offdiagonal block to be examined in each row in order to avoid blank blocks.

IFR : Specifies whether any boundary conditions have to be applied to the diagonal block of row 'I' or not.

5. Subroutine SOLVE

KMM : Same as MM1 in MERGE.

IFLG : Indicates whether block AK2 is singular or nearly singular.

6. Subroutine DISP

JLL : Indicates the number of non blank elements in the last load vector

7. Subroutine STRESS

CORDG(NPEL,3) : Vector of element nodal coordinates in the local system.

SNELM(NPEL,6) : Array of element nodal strains.

SSELM(NPEL,6) : Array of element nodal stresses.

SSNN(6) : Strain vector for one node.

SSSS(6) : Stress vector for one node.

UEL(NST) : Vector of element nodal displacements.

APPENDIX B

DIRECT ACCESS DATA FILES REQUIREMENTS

In this Appendix the direct access data file requirements as well as the necessary control cards for use of the compiler of release 18 of an IBM 360 operating system are described.

A. DATA FILE SPACE REQUIREMENTS

The data file space requirements are described by the variables NRX, LRBX, LRWX as they are explained in Section A of Appendix A.

The required values of these variables are as follows: (the values of LRWX are not presented since always LRWX= LRBX/4)

NR7 = MM x NN x NREC7

LRB7 = NCOUNT x 8/NREC7

NR8 = NEL x 4 for quadratic elements

NEL x 12 for cubic elements

LRB8 = 7200 for quadratic elements

6144 for cubic elements

NR9 = NEL

LRB9 = 80 for quadratic elements

128 for cubic elements

NR10 = NDPT

LRB10 = 24

NR12 = NDPT

LRB12 = 48

NR13 = NDPT

LRB13 = 48

NR14 = NCLD

LRB14 = 32

NR15 = NPBC

LRB15 = 16

NR16 = NEL

LRB16 = 4

NR17 = NPBC

LRB17 = 32

NR19 = NEL

LRB19 = 480

NR20 = NN

LRB20 = NS x 8

NR21 = NN x NS/NDF

LRB21 = 24

NR22 = NPBC2

LRB22 = 24

B. CONTROL CARDS

Job control language cards for an IBM 360 operating system are of the form:

```
GO.FTXXF001 DD UNIT = SYSDA, SPACE = (LRBX, (NRX,1)),DISP=
    (NEW,DELETE)
```

where X and XX represent the number of the corresponding file.

These cards may be produced by the auxiliary computer program named JCL and whose listing is presented in Appendix H. This program is capable of printing and punching the JCL cards as well as the data files characteristics (Section IV.C.1) for a series of problems in one run. One data card is required for each problem. This card must contain the variables: NEL, NPEL, MM, NS, NDPT, NPBC, NPBC2, NCLD, NREC7, NMAT punched in format (1X,1015). Following the data card of the last problem, a card with a negative integer punched in columns one to six right justified, must be given. The control cards required for this program in order to be used in an IBM 360 OS are shown in Appendix F.

In case the value of MM is not known, it can be computed by the program from connectivity data. In this case MM is given the value zero in the data card described above and which must be immediately followed by the data cards:

1. Object Time Format (20A4)

   Number of cards: one

   The format under which connectivity data will be read is punched. This format must provide for reading NPEL integers at a time, representing one row of the connectivity matrix.

2. Connectivity Data

   Number of cards: NEL x (number of cards per row of connectivity matrix as per specified format.)

   Each individual row of the connectivity matrix is punched in sequence under the specified format.

The input data of program JCL are printed in the output for checking purposes. A sample output is presented in Appendix E.

C. DEFINE FILE STATEMENTS

The DEFINE FILE statements contained in subroutine DISK and defining the space requirements for random access files must be adjusted to the needs of the larger problem of each run. The form of these statements is:

```
DEFINE FILE X(NRX,LRWX,U,IX)
```

where X represents the number of the corresponding file. These statements are also produced by program JCL as described in Section B of this Appendix.

```
        16       7200       1800          4          8       7200       1800
         2         80         32         24          6         32         56
        14         32         56         14          3         32          6
        16          0          0          6         32          2        480
         2        480        120         40         24          6          0
         0
START
1  SIMPLY SUPPORTED BEAM - QUADRATIC ELEMENTS - MESH 1X1X2
     2    32    60    20     1     3     6          60     1
  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T    CONN. MTR.
(1X,20I3)
   6  4  1  2  3  5  8  7 11  9 10 12 18 16 13 14 15 17 20 19     1
  18 16 13 14 15 17 20 19 23 21 22 24 30 28 25 26 27 29 32 31     1
 COORD.      JOINT          X                Y                Z          KND
(10X,I6,3F18.5,I5)
                 1          0.0              2.00000          24.0
                 2          0.0              1.00000          24.0
                 3          0.0              0.0              24.0
                 4          1.00000          2.00000          24.0
                 5          1.00000          0.0              24.0
                 6          2.00000          2.00000          24.0
                 7          2.00000          1.00000          24.0
                 8          2.00000          0.0              24.0
                 9          0.0              2.00000          18.0
                10          0.0              0.0              18.0
                11          2.00000          2.00000          18.0
                12          2.00000          0.0              18.0
                13          0.0              2.00000          12.0
                14          0.0              1.00000          12.0
                15          0.0              0.0              12.0
                16          1.00000          2.00000          12.0
                17          1.00000          0.0              12.0
                18          2.00000          2.00000          12.0
                19          2.00000          1.00000          12.0
                20          2.00000          0.0              12.0
                21          0.0              2.00000           6.0
                22          0.0              0.0               6.0
                23          2.00000          2.00000           6.0
                24          2.00000          0.0               6.0
                25          0.0              2.00000           0.0
                26          0.0              1.00000           0.0
                27          0.0              0.0               0.0
                28          1.00000          2.00000           0.0
                29          1.00000          0.0               0.0
                30          2.00000          2.00000           0.0
                31          2.00000          1.00000           0.0
                32          2.00000          0.0               0.0
 NMAT     Y.MOD.    P.RATIO     ALPHA                    ELEMENT DATA
```

```
(1X,1I3,1X,1F10.0,2F10.5)
  1 30000000.00.3
    N.PT      XLD          YLD          ZLD      CONCENTRATED LOAD DATA
(5X,1I5,3F10.5)
         13              -16.0
         16              -64.0
         18              -16.0
       JT              X            Y            Z          BOUNDARY CONDITIONS
(4I10)
          3              1            1
          5              1            1
          8              1            1
         27              1            1            1
         29              1            1            1
         32              1            1            1
START
  END OF PROBLEM
   -1
```

```
        72       6144       1536         12         24       6144       1536
         2        128         52         24          6         52         56
        14         52         56         14          4         32          8
        16          0          0          8         32          2        480
         2        768        192         64         24          6          0
         0
START
1  SIMPLY SUPPORTED BEAM - CUBIC ELEMENTS - MESH 1X1X2
     2    52    96    32     1     4     8          60     1
        A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P    CONN. MTR.
        Q   R   S   T   U   V   0   0   0   0  A1  B1  C1  D1  E1  F1
(5X,16I3)
        9   7   5   1   2   3   4   6   8  12  11  10  15  13  14  16
       19  17  18  20  29  27  25  21  22  23  24  26  28  32  31  30
       29  27  25  21  22  23  24  26  28  32  31  30  35  33  34  36
       39  37  38  40  49  47  45  41  42  43  44  46  48  52  51  50
 COORD.        JOINT              X                    Y                    Z                KND
(10X,I6,3F18.5,I5)
                    1          0.0                2.00000              24.00000
                    2          0.0                1.33333              24.00000
                    3          0.0                0.66667              24.00000
                    4          0.0                0.0                  24.00000
                    5          0.66667            2.00000              24.00000
                    6          0.66667            0.0                  24.00000
                    7          1.33333            2.00000              24.00000
                    8          1.33333            0.0                  24.00000
                    9          2.00000            2.00000              24.00000
                   10          2.00000            1.33333              24.00000
                   11          2.00000            0.66667              24.00000
                   12          2.00000            0.0                  24.00000
                   13          0.0                2.00000              20.00000
                   14          0.0                0.0                  20.00000
                   15          2.00000            2.00000              20.00000
                   16          2.00000            0.0                  20.00000
                   17          0.0                2.00000              16.00000
                   18          0.0                0.0                  16.00000
                   19          2.00000            2.00000              16.00000
                   20          2.00000            0.0                  16.00000
                   21          0.0                2.00000              12.00000
                   22          0.0                1.33333              12.00000
                   23          0.0                0.66667              12.00000
                   24          0.0                0.0                  12.00000
                   25          0.66667            2.00000              12.00000
                   26          0.66667            0.0                  12.00000
                   27          1.33333            2.00000              12.00000
                   28          1.33333            0.0                  12.00000
                   29          2.00000            2.00000              12.00000
                   30          2.00000            1.33333              12.00000
```

```
                31          2.00000          0.66667         12.00000
                32          2.00000          0.0             12.00000
                33          0.0              2.00000          8.00000
                34          0.0              0.0              8.00000
                35          2.00000          2.00000          8.00000
                36          2.00000          0.0              8.00000
                37          0.0              2.00000          4.00000
                38          0.0              0.0              4.00000
                39          2.00000          2.00000          4.00000
                40          2.00000          0.0              4.00000
                41          0.0              2.00000          0.0
                42          0.0              1.33333          0.0
                43          0.0              0.66667          0.0
                44          0.0              0.0              0.0
                45          0.66667          2.00000          0.0
                46          0.66667          0.0              0.0
                47          1.33333          2.00000          0.0
                48          1.33333          0.0              0.0
                49          2.00000          2.00000          0.0
                50          2.00000          1.33333          0.0
                51          2.00000          0.66667          0.0
                52          2.00000          0.0              0.0
 NMAT    Y.MOD.    P.RATIO       ALPHA                    ELEMENT DATA
(1X,1I3,1X,1F10.0,2F10.5)
   1 30000000.00.3
    N.PT       XLD         YLD         ZLD     CONCENTRATED LOAD DATA
(5X,1I5,3F10.5)
        21          -12.0
        25          -36.0
        27          -36.0
        29          -12.0
      JT            X           Y           Z          BOUNDARY CONDITIONS
(4I10)
         4          1           1
         6          1           1
         8          1           1
        12          1           1
        44          1           1           1
        46          1           1           1
        48          1           1           1
        52          1           1           1
START
   -1
```

APPENDIX D

SAMPLE OUTPUT FOR TRISOP

A sample output for the solution of a problem by use of quadratic elements is presented in this Appendix. The problem chosen is that of a simply supported beam with a line load acting on the mid-section and discretized in a 1x1x2 mesh. This problem is one of those discussed in Section V.

The form of the output for a solution using cubic elements will be identical except for the connectivity matrix. The form of the connectivity matrix for the same problem solved by use of cubic elements will be as follows:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	Y	Z	A1	B1	C1	D1	E1	F1	G1
9	7	5	1	2	3	4	6	8	12	11	10	15	13	14	16
19	17	18	20	29	27	25	21	22	23	24	26	28	32	31	30
29	27	25	21	22	23	24	26	28	32	31	30	35	33	34	36
39	37	38	40	49	47	45	41	42	43	44	46	48	52	51	50

SIMPLY SUPPORTED BEAM - QUADRATIC ELEMENTS - MESH 1X1X2

```
   NEL....TOTAL NUMBER OF ELEMENTS.....................  2
   NDPT...TOTAL NUMBER OF NODAL POINTS.................. 32
   NMAT...NUMBER OF DIFFERENT MATERIALS.................  1
          (MAXIMUM IS 10)
   NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER..... 60
   NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND.......  6
   NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL.....  0
   NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD.........  3
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	CONN. MTR.
6	4	1	2	3	5	8	7	11	9	10	12	18	16	13	14	15	17	20	19	
18	16	13	14	15	17	20	19	23	21	22	24	30	28	25	26	27	29	32	31	

```
   NEQ....TOTAL NUMBER OF EQUATIONS.....................   96
   NBAND..HALF-BAND WIDTH OF THE SYSTEM.................   60
   NN.....NUMBER OF BLOCKS PER COLUMN...................    2
   MM.....NUMBER OF BLOCKS PER ROW......................    2
   NCOUNT.NUMBER OF COEFFICIENTS PER BLOCK.............. 3600
   NSTE...NUMBER OF COEFFICIENTS PER ELEMENT............ 3600
```

COORD. JOINT	X	Y	Z	KND
1	0.0	2.00000	24.00000	0
2	0.0	1.00000	24.00000	0
3	0.0	0.0	24.00000	0
4	1.00000	2.00000	24.00000	0
5	1.00000	0.0	24.00000	0
6	2.00000	2.00000	24.00000	0
7	2.00000	1.00000	24.00000	0
8	2.00000	0.0	24.00000	0
9	0.0	2.00000	18.00000	0
10	0.0	0.0	18.00000	0
11	2.00000	2.00000	18.00000	0
12	2.00000	0.0	18.00000	0
13	0.0	2.00000	12.00000	0
14	0.0	1.00000	12.00000	0
15	0.0	0.0	12.00000	0
16	1.00000	2.00000	12.00000	0
17	1.00000	0.0	12.00000	0
18	2.00000	2.00000	12.00000	0
19	2.00000	1.00000	12.00000	0
20	2.00000	0.0	12.00000	0
21	0.0	2.00000	6.00000	0
22	0.0	0.0	6.00000	0
23	2.00000	2.00000	6.00000	0
24	2.00000	0.0	6.00000	0
25	0.0	2.00000	0.0	0
26	0.0	1.00000	0.0	0
27	0.0	0.0	0.0	0
28	1.00000	2.00000	0.0	0
29	1.00000	0.0	0.0	0
30	2.00000	2.00000	0.0	0
31	2.00000	1.00000	0.0	0
32	2.00000	0.0	0.0	0

NMAT	Y.MOD.	P.RATIO	ALPHA	ELEMENT DATA
1	30000000.	0.30000	0.0	

```
N.PT      XLD         YLD         ZLD      CONCENTRATED LOAD DATA

    13    0.0      -16.00000     0.0
    16    0.0      -64.00000     0.0
    18    0.0      -16.00000     0.0
   JT             X           Y           Z          BOUNDARY CONDITIONS

     3             1           1           0
     5             1           1           0
     9             1           1           0
    27             1           1           1
    29             1           1           1
    32             1           1           1
***** INPUT *****
 TIME =   14.62 SECONDS(CPU =    0.80 SECONDS)

***** SORT *****
 TIME =    5.86 SECONDS(CPU =    0.26 SECONDS)

***** ESTF *****
 TIME =   27.59 SECONDS(CPU =   15.61 SECONDS)

***** MERGE *****
 TIME =   17.20 SECONDS(CPU =    3.21 SECONDS)

***** FLOAD *****
 TIME =    5.50 SECONDS(CPU =    0.94 SECONDS)

***** BCOND *****
 TIME =    4.63 SECONDS(CPU =    1.20 SECONDS)

CONDITION NUMBER          2.42587908136393400 04        2.80803418803418800 09    8.63905109026544400-06
CONDITION NUMBER          2.43729213921573700 04        1.00133957931525000 09    2.43399460339160300-05
***** SOLVE *****
 TIME =   39.82 SECONDS(CPU =   27.11 SECONDS)
```

D-I-S-P-L-A-C-E-M-E-N-T-S

N.PT	X	Y	Z
1	-1.6108801503157410D-06	-1.5509006246076870D-06	3.4463609617350360D-07
2	-2.2712961766677040D-07	-1.7404130631518820D-06	8.6210668881769490D-05
3	0.0	0.0	1.7216516913280500D-04
4	-5.1971106753842560D-18	-6.0533955609158370D-07	2.7171956807745680D-07
5	0.0	0.0	1.7233551034776710D-04
6	1.6108801503061410D-06	-1.5509006246023690D-06	3.4463609614478850D-07
7	2.2712961766296360D-07	-1.7404130631468130D-06	8.6210668881740870D-05
8	0.0	0.0	1.7216516913277540D-04
9	-2.2044106492738100D-06	-4.1726844920351770D-04	3.7560894572957530D-05
10	2.1734037971615410D-06	-4.1730929743614670D-04	1.3501378089229570D-04
11	2.2044106491086170D-06	-4.1726844920348510D-04	3.7560894572945860D-05
12	-2.1734037972562100D-06	-4.1730929743611700D-04	1.3501378089228370D-04
13	-2.7441674906415220D-06	-5.7605369050687150D-04	8.6125608598045380D-05
14	-3.1430545612939840D-07	-5.7705063704974770D-04	8.6143592940970110D-05
15	2.6969539757900240D-06	-5.7558443897449950D-04	8.6161310071790280D-05
16	-1.1556001604618190D-16	-5.7476005675415500D-04	8.6143619060498580D-05
17	-2.7180053985498600D-17	-5.7391247608632370D-04	8.6119093034936250D-05
18	2.7441674904087690D-06	-5.7605869050677370D-04	8.6125608598053300-05
19	3.1430545598618690D-07	-5.7705063704965430D-04	8.6143592940977850D-05
20	-2.6969539758441970D-06	-5.7558443897440940D-04	8.6161310071798120D-05
21	-2.2048355882337200D-06	-4.1725670989150350D-04	1.3472478764208340D-04
22	2.1664590091669740D-06	-4.1729580187759630D-04	3.7231243742973040D-05
23	2.2048355881172600D-06	-4.1725670989146090D-04	1.3472478764209450D-04
24	-2.1664590019500300D-06	-4.1729580187755280D-04	3.7231243742985370D-05
25	-1.6134527132876710D-06	-1.5499356704145230D-06	1.7197492955401010D-04
26	-2.2367525602319530D-07	-1.7402851670226140D-06	8.6031367206855560D-05
27	0.0	0.0	0.0
28	-2.2579895150190660D-18	-6.0351476868240920D-07	1.7197518870486640D-04
29	0.0	0.0	0.0
30	1.6134527132832660D-06	-1.5499356704093550D-06	1.7197492955402260D-04
31	2.2367525602141480D-07	-1.7402851670191110D-06	8.6031367206863730D-05
32	0.0	0.0	0.0

R-E-A-C-T-I-O-N-S

N.PT	X	Y	Z
2	2.3658853372884870D 01	-1.5091973755734090D 01	0.0
5	2.6762408448878810D-11	-1.7816052489112260D 01	0.0
8	-2.3658853729456610D 01	-1.5091973755648660D 01	0.0
27	2.4127365073037160D 01	-1.4815110525534580D 01	4.5190459456394300D 00
29	1.4322801094607820D-10	-1.8369778949577750D 01	-9.0380918915346030D 00
32	-2.4127365073164580D 01	-1.4815110525420530D 01	4.5190459457794850D 00

EQUILIBRIUM CHECK

-1.8189894035458560D-11 -9.6000000010278700 01 -1.1568812574580530D-10

***** DISP *****

TIME = 11.36 SECONDS(CPU = 0.82 SECONDS)

S-T-R-A-I-N-S / S-T-R-E-S-S-E-S FOR ELEMENT 1

JOINT	SNX/SSX	SNY/SSY	SNZ/SSZ	SNXY/SSXY	SNYZ/SSYZ	SNZX/SSZX
6	1.610880D-06	1.154475D-06	-5.257005D-06	7.093886D-08	4.875068D-06	4.243017D-08
6	-5.950550D 00	-1.648297D 01	-1.644402D 02	8.185253D-01	5.625078D 01	4.895789D-01
4	1.610880D-06	1.627256D-06	-5.264582D-06	-9.330200D-19	4.724016D-06	2.175485D-18
4	2.101046D 00	2.478944D 00	-1.565635D 02	-1.076562D-11	5.450788D 01	2.510175D-11
1	1.610880D-06	1.154475D-06	-5.257005D-06	-7.093886D-08	4.875068D-06	-4.243017D-08
1	-5.950550D 00	-1.648297D 01	-1.644402D 02	-8.185253D-01	5.625078D 01	-4.895789D-01
2	2.271296D-07	-7.754503D-07	-2.379592D-08	1.401210D-07	5.029041D-06	-2.831732D-08
2	-4.660566D 00	-2.779703D 01	-1.045116D 01	1.616781D 00	5.802739D 01	-3.267383D-01
3	-2.365885D-19	-2.705376D-06	5.216808D-06	3.511808D-07	5.138995D-06	-1.590393D-07
3	4.346709D 01	-1.896466D 01	1.638550D 02	4.052087D 00	5.929610D 01	-1.835069D 00
5	-2.365885D-19	-2.232595D-06	5.234521D-06	-1.605445D-18	4.869703D-06	-1.253741D-18
5	5.195641D 01	4.349773D-01	1.727530D 02	-1.552437D-11	5.618888D 01	-1.446624D-11
8	-2.365885D-19	-2.705376D-06	5.216808D-06	-3.511808D-07	5.138995D-06	1.590393D-07
8	4.346709D 01	-1.896466D 01	1.638550D 02	-4.052087D 00	5.929610D 01	1.835069D 00
7	2.271296D-07	-7.754503D-07	-2.379592D-08	-1.401210D-07	5.029041D-06	2.831732D-08
7	-4.660566D 00	-2.779703D 01	-1.045116D 01	-1.616781D 00	5.802739D 01	3.267383D-01
11	2.204411D-06	2.214459D-06	-7.148414D-06	2.323242D-07	-8.066939D-07	-3.953355D-08
11	3.628898D 00	3.860787D 00	-2.122055D 02	2.680663D 00	-9.308006D 00	-4.561563D-01
9	2.204411D-06	2.214459D-06	-7.148414D-06	-2.323242D-07	-8.066939D-07	3.953355D-08
9	3.628898D 00	3.860787D 00	-2.122055D 02	-2.680663D 00	-9.308006D 00	4.561563D-01
10	-2.173404D-06	-2.173611D-06	7.166988D-06	-1.293326D-07	-8.051734D-07	-9.662199D-08
10	-1.348238D 00	-1.353019D 00	2.141993D 02	-1.492299D 00	-9.290462D 00	-1.114869D 00
12	-2.173404D-06	-2.173611D-06	7.166988D-06	1.293326D-07	-8.051734D-07	9.662199D-08
12	-1.348238D 00	-1.353019D 00	2.141993D 02	1.492299D 00	-9.290462D 00	1.114869D 00
18	2.744167D-06	2.221019D-06	-9.039824D-06	-4.581042D-07	5.036313D-06	-1.214973D-07
18	-7.195626D 00	-1.926829D 01	-2.791339D 02	-5.285817D 00	5.811131D 01	-1.401891D 00
16	2.744167D-06	1.984354D-06	-9.047401D-06	4.587787D-18	5.037004D-06	5.932619D-18
16	-1.142289D 01	-2.895734D 01	-2.835360D 02	5.305139D-11	5.811928D 01	6.845329D-11
13	2.744167D-06	2.221019D-06	-9.039824D-06	4.581042D-07	5.036313D-06	1.214973D-07
13	-7.195626D 00	-1.926829D 01	-2.791339D 02	5.285817D 00	5.811131D 01	1.401891D 00
14	3.143055D-07	-2.371258D-07	3.497525D-08	3.500359D-07	4.927879D-06	1.160637D-07
14	9.194346D 00	-3.530990D 00	2.748264D 00	4.038876D 00	5.686015D 01	1.339197D 00
15	-2.696954D-06	-2.695270D-06	9.117169D-06	2.419676D-07	4.775427D-06	-3.420464D-08
15	2.232791D 00	2.271643D 00	2.748664D 02	2.791934D 00	5.510108D 01	-3.946689D-01
17	-2.696954D-06	-2.931935D-06	9.134882D-06	1.017617D-18	4.657877D-06	-5.072033D-18
17	-1.556752D 00	-6.979390D 00	2.714856D 02	1.174174D-11	5.374473D 01	-5.852346D-11
20	-2.696954D-06	-2.695270D-06	9.117169D-06	-2.419676D-07	4.775427D-06	3.420464D-08
20	2.232791D 00	2.271643D 00	2.748664D 02	-2.791934D 00	5.510108D 01	3.946689D-01
19	3.143055D-07	-2.371258D-07	3.497525D-08	-3.500359D-07	4.927879D-06	-1.160637D-07
19	9.194346D 00	-3.530990D 00	2.748264D 00	-4.038876D 00	5.686015D 01	-1.339197D 00

S-T-R-A-I-N-S / S-T-R-E-S-S-E-S FOR ELEMENT 2

JOINT	SNX/SSX	SNY/SSY	SNZ/SSZ	SNXY/SSXY	SNYZ/SSYZ	SNZX/SSZX
18	2.744167D-06	2.2210190-06	-9.0456160-06	-4.581042D-07	-5.0763820-06	4.9522140-08
18	-7.295884D 00	-1.9368540 01	-2.7936780 02	-5.2858170 00	-5.8573640 01	5.7152470-01
16	2.744167D-06	1.9843540-06	-9.0441370-06	4.5977870-18	-5.0169180-06	-5.7259030-10
16	-1.136640D 01	-2.9900550 01	-2.8340420 02	5.3051390-11	-5.7887510 01	-6.6068110-11
13	2.744167D-06	2.2210190-06	-9.0456160-06	4.5810420-07	-5.0763820-06	-4.9532140-08
13	-7.295884D 00	-1.9358540 01	-2.7936780 02	5.2858170 00	-5.8573640 01	-5.7152470-01
14	3.1430550-07	-2.3712580-07	3.8500930-08	3.5003590-07	-4.9677170-06	-1.6375280-07
14	9.2553670 00	-3.4699680 00	2.8906480 00	4.0388760 00	-5.7319810 01	-1.8894560 00
15	-2.6969540-06	-2.6952700-06	9.1299130-06	2.4196760-07	-4.8150930-06	-1.3234860-07
15	2.4533640 00	2.4922160 00	2.7538110 02	2.7919340 00	-5.5558760 01	-1.5270990 00
17	-2.6969540-06	-2.9319350-06	9.1263550-06	1.0176170-18	-4.6373160-06	1.8151920-10
17	-1.703643D 00	-7.1262810 00	2.7114290 02	1.1741740-11	-5.3507490 01	2.0944520-11
20	-2.6969540-06	-2.6952700-06	9.1299130-06	-2.4196760-07	-4.8150930-06	1.3234860-07
20	2.4533640 00	2.4922160 00	2.7538110 02	-2.7919340 00	-5.5558760 01	1.5270990 00
19	3.1430550-07	-2.3712580-07	3.8500930-08	-3.5003590-07	-4.9677170-06	1.6375280-07
19	9.2553670 00	-3.4699680 00	2.8906480 00	-4.0388760 00	-5.7319810 01	1.8894560 00
23	2.2048360-06	2.2139360-06	-7.1541100-06	2.3294500-07	8.2700630-07	7.5957620-08
23	3.5384180 00	3.7484190 00	-2.1243730 02	2.6878270 00	9.5423810 00	8.7643410-01
21	2.2048360-06	2.2139360-06	-7.1541100-06	-2.3294500-07	8.2700630-07	-7.5957620-08
21	3.5384180 00	3.7484190 00	-2.1243730 02	-2.6878270 00	9.5423810 00	-8.7643410-01
22	-2.1664590-06	-2.1748440-06	7.1801090-06	-1.2133200-07	8.2543810-07	1.8252910-07
22	-8.6201820-01	-1.0555100 00	2.1482800 02	-1.3999850 00	9.5242850 00	2.1061050 00
24	-2.1664590-06	-2.1748440-06	7.1801090-06	1.2133200-07	8.2543810-07	-1.8252910-07
24	-8.6201820-01	-1.0555100 00	2.1482800 02	1.3999850 00	9.5242850 00	-2.1061050 00
30	1.6134530-06	1.1556670-06	-5.2626040-06	7.9986750-08	-4.7935350-06	1.0238310-07
30	-5.9229360 00	-1.6487230 01	-1.6460120 02	9.2292410-01	-5.5310020 01	1.1813430 00
28	1.6134530-06	1.6288770-06	-5.2611250-06	1.0335010-18	-4.7640550-06	-2.8372540-10
28	2.2928470 00	2.6487980 00	-1.5635120 02	1.2501930-11	-5.4969860 01	-3.2760630-11
25	1.6134530-06	1.1556670-06	-5.2626040-06	-7.9986750-08	-4.7935350-06	-1.0238310-07
25	-5.9229360 00	-1.6487230 01	-1.6460120 02	-9.2292410-01	-5.5310020 01	-1.1813430 00
26	2.2367530-07	-7.7496780-07	-1.9796640-08	1.3969450-07	-4.9477280-06	1.2469940-07
26	-4.7225000 00	-2.7768110 01	-1.0341080 01	1.6118600 00	-5.7089170 01	1.4388390 00
27	-2.4127370-10	-2.7056020-06	5.2303050-06	3.5937580-07	-5.0579610-06	4.9740680-07
27	4.3696780 01	-1.8740200 01	1.6439610 02	4.1466440 00	-5.8361090 01	5.7393100 00
29	-2.4127370-10	-2.2323920-06	5.2267870-06	-7.0698130-10	-4.9101680-06	-2.4219320-10
29	5.1826070 01	3.0933250-01	1.7244420 02	-8.1574770-12	-5.6655780 01	-2.7945370-11
32	-2.4127370-10	-2.7056020-06	5.2303050-06	-3.5937580-07	-5.0579610-06	-4.9740680-07
32	4.3696780 01	-1.8740200 01	1.6439610 02	-4.1466440 00	-5.8361090 01	-5.7393100 00
31	2.2367530-07	-7.7496780-07	-1.9796640-08	-1.3969450-07	-4.9477280-06	-1.2469940-07
31	-4.7225000 00	-2.7768110 01	-1.0341080 01	-1.6118600 00	-5.7089170 01	-1.4388390 00

A-V-E-R-A-G-E S-T-R-E-S-S-E-S AT THE JOINTS

JOINT	SSX	SSY	SSZ	SSXY	SSYZ	SSZX
1	-5.9505500 00	-1.6482970 01	-1.6444020 02	-8.1852530-01	5.6250780 01	-4.8957890-01
2	-4.6605660 00	-2.7797030 01	-1.0451160 01	1.6167810 00	5.8027390 01	-3.2673830-01
3	4.3467090 01	-1.8964660 01	1.6385500 02	4.0520870 00	5.9296100 01	-1.8350690 00
4	2.1010460 00	2.4783440 00	-1.5656250 02	-1.0765620-11	5.4507880 01	2.5101750-11
5	5.1956410 01	4.3497730-01	1.7275300 02	-1.8524370-11	5.6188980 01	-1.4466240-11
6	-5.9505500 00	-1.6482970 01	-1.6444020 02	8.1852530-01	5.6250780 01	4.8957890-01
7	-4.6605660 00	-2.7797030 01	-1.0451160 01	-1.6167810 00	5.8027390 01	3.2673830-01
8	4.3467090 01	-1.8964660 01	1.6385500 02	-4.0520870 00	5.9296100 01	1.8350690 00
9	3.6288980 00	3.9607870 00	-2.1220550 02	-2.6806630 00	-9.3080060 00	4.5615630-01
10	-1.3482380 00	-1.3530190 00	2.1419930 02	-1.4922990 00	-9.2904620 00	-1.1148690 00
11	3.6288980 00	3.8607870 00	-2.1223550 02	2.6806630 00	-9.3080060 00	-4.5615630-01
12	-1.3482380 00	-1.3530190 00	2.1419930 02	1.4922990 00	-9.2904620 00	1.1148690 00
13	-7.2457550 00	-1.9318420 01	-2.7925080 02	5.2858170 00	-2.3116490-01	4.1518340-01
14	9.2248570 00	-3.5004790 00	2.8194560 00	4.0388760 00	-2.2983190-01	-2.7512950-01
15	2.3430780 00	2.3819300 00	2.7512370 02	2.7919340 00	-2.2883980-01	-9.6088400-01
16	-1.1394640 01	-2.8928790 01	-2.8347010 02	5.3051390-11	1.1588230-01	1.1925910-12
17	-1.6301980 00	-7.0528360 00	2.7131420 02	1.1741740-11	1.1862080-01	-1.8789470-11
18	-7.2457550 00	-1.9318420 01	-2.7925080 02	-5.2858170 00	-2.3116490-01	-4.1518340-01
19	9.2248570 00	-3.5004790 00	2.8194560 00	-4.0388760 00	-2.2983190-01	2.7512950-01
20	2.3430780 00	2.3819300 00	2.7512370 02	-2.7919340 00	-2.2883980-01	9.6088400-01
21	3.5384180 00	3.7484190 00	-2.1243730 02	-2.6878270 00	9.5423810 00	-8.7643410-01
22	-8.6201820-01	-1.0555100 00	2.1482800 02	-1.3999850 00	9.5242850 00	2.1061050 00
23	3.5384180 00	3.7484190 00	-2.1243730 02	2.6878270 00	9.5423810 00	8.7643410-01
24	-8.6201820-01	-1.0555100 00	2.1482800 02	1.3999850 00	9.5242850 00	-2.1061050 00
25	-5.9229360 00	-1.6487230 01	-1.6460120 02	-9.2292410-01	-5.5310020 01	-1.1813430 00
26	-4.7225000 00	-2.7768110 01	-1.0341080 01	1.6118600 00	-5.7089170 01	1.4388390 00
27	4.3696780 01	-1.8740200 01	1.6439610 02	4.1466440 00	-5.8361090 01	5.7393100 00
28	2.2928470 00	2.6497980 00	-1.5635120 02	1.2501930-11	-5.4969860 01	-3.2760630-11
29	5.1926070 01	3.0933250-01	1.7244420 02	-9.1574770-12	-5.6655780 01	-2.7945370-11
30	-5.9229360 00	-1.6487230 01	-1.6460120 02	9.2292410-01	-5.5310020 01	1.1813430 00
31	-4.7225000 00	-2.7768110 01	-1.0341080 01	-1.6118600 00	-5.7089170 01	-1.4388390 00
32	4.3696780 01	-1.8740200 01	1.6439610 02	-4.1466440 00	-5.8361090 01	-5.7393100 00

A-V-E-R-A-G-E S-T-R-A-I-N-S AT THE JOINTS

JOINT	SNX	SNY	SNZ	SNXY	SNYZ	SNZX
1	1.610880D-06	1.154475D-06	-5.257005D-06	-7.093886D-08	4.875068D-06	-4.243017D-08
2	2.271296D-07	-7.754503D-07	-2.379592D-08	1.401210D-07	5.029041D-06	-2.831732D-08
3	-2.365985D-19	-2.705376D-06	5.216908D-06	3.511908D-07	5.138995D-06	-1.590393D-07
4	1.610880D-06	1.627256D-06	-5.264582D-06	-8.330203D-19	4.724016D-06	2.175485D-18
5	-2.365985D-19	-2.232595D-06	5.234521D-06	-1.605445D-18	4.869703D-06	-1.253741D-18
6	1.610880D-06	1.154475D-06	-5.257005D-06	7.093886D-08	4.875068D-06	4.243017D-08
7	2.271296D-07	-7.754503D-07	-2.379592D-08	-1.401210D-07	5.029041D-06	2.831732D-08
8	-2.365985D-19	-2.705376D-06	5.216908D-06	-3.511808D-07	5.138995D-06	1.590393D-07
9	2.204411D-06	2.214459D-06	-7.148414D-06	-2.323242D-07	-8.066939D-07	3.953355D-08
10	-2.173404D-06	-2.173611D-06	7.166988D-06	-1.293326D-07	-8.051734D-07	-9.662199D-08
11	2.204411D-06	2.214459D-06	-7.148414D-06	2.323242D-07	-8.066939D-07	-3.953355D-08
12	-2.173404D-06	-2.173611D-06	7.166988D-06	1.293326D-07	-8.051734D-07	9.662199D-08
13	2.744167D-06	2.221019D-06	-9.042720D-06	4.581042D-07	-2.003429D-08	3.598256D-08
14	3.143055D-07	-2.371258D-07	3.673809D-08	3.500359D-07	-1.991877D-08	-2.384455D-08
15	-2.696954D-06	-2.695270D-06	9.123541D-06	2.419676D-07	-1.983278D-08	-8.327661D-08
16	2.744167D-06	1.984354D-06	-9.045769D-06	4.597787D-18	1.004313D-08	1.033579D-19
17	-2.696954D-06	-2.931935D-06	9.130628D-06	1.017617D-18	1.028047D-08	-1.628421D-18
18	2.744167D-06	2.221019D-06	-9.042720D-06	-4.581042D-07	-2.003429D-08	-3.598256D-08
19	3.143055D-07	-2.371258D-07	3.673809D-08	-3.500359D-07	-1.991877D-08	2.384455D-08
20	-2.696954D-06	-2.695270D-06	9.123541D-06	-2.419676D-07	-1.983278D-08	8.327661D-08
21	2.204836D-06	2.213936D-06	-7.154110D-06	-2.329450D-07	8.270063D-07	-7.595762D-08
22	-2.166459D-06	-2.174844D-06	7.180109D-06	-1.213320D-07	8.254381D-07	1.825291D-07
23	2.204836D-06	2.213936D-06	-7.154110D-06	2.329450D-07	8.270063D-07	7.595762D-08
24	-2.166459D-06	-2.174844D-06	7.180109D-06	1.213320D-07	8.254381D-07	-1.825291D-07
25	1.613453D-06	1.155667D-06	-5.262604D-06	-7.998675D-08	-4.793535D-06	-1.023831D-07
26	2.236753D-07	-7.749678D-07	-1.979664D-08	1.396945D-07	-4.947728D-06	1.246994D-07
27	-2.412737D-19	-2.705602D-06	5.230305D-06	3.593758D-07	-5.057961D-06	4.974068D-07
28	1.613453D-06	1.628877D-06	-5.261125D-06	1.083501D-18	-4.764055D-06	-2.839254D-18
29	-2.412737D-19	-2.232392D-06	5.226787D-06	-7.069813D-19	-4.910168D-06	-2.421932D-18
30	1.613453D-06	1.155667D-06	-5.262604D-06	7.998675D-08	-4.793535D-06	1.023831D-07
31	2.236753D-07	-7.749678D-07	-1.979664D-08	-1.396945D-07	-4.947728D-06	-1.246994D-07
32	-2.412737D-19	-2.705602D-06	5.230305D-06	-3.593758D-07	-5.057961D-06	-4.974068D-07

***** STRESS *****
TIME = 32.78 SECONDS(CPU = 3.05 SECONDS)

END OF PROBLEM

APPENDIX E

SAMPLE OUTPUTS FOR AUXILIARY PROGRAM JCL

Two different sample outputs for the auxiliary program JCL are presented in this Appendix, one for a problem utilizing quadratic elements and one for a problem utilizing cubic elements. Both problems are from those presented in Section V for a simply supported beam.

```
            ****** INPUT DATA ******


NEL....TOTAL NUMBER OF ELEMENTS.....................   4
NDPT...TOTAL NUMBER OF NODAL POINTS.................  56
NMAT...NUMBER OF DIFFERENT MATERIALS................   1
       (MAXIMUM IS  10)
NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER.....  60
NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND.....   6
NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL....   0
NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD.......   2
NN.....NUMBER OF BLOCKS PER COLUMN..................   3
MM.....NUMBER OF BLOCKS PER ROW.....................   2
NPEL...NUMBER OF NODES PER ELEMENT..................  20
NREC7..NUMBER OF RECORDS OF FILE NO 7...............   4



            ****** JCL CARDS ******


//GO.FT07F001 DD UNIT=SYSDA,SPACE=(7200,(0024,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(7200,(0016,1)),DISP=(NEW,DELETE)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(0080,(0004,1)),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(0024,(0056,1)),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=SYSDA,SPACE=(0056,(0056,1)),DISP=(NEW,DELETE)
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(0056,(0056,1)),DISP=(NEW,DELETE)
//GO.FT20F001 DD UNIT=SYSDA,SPACE=(0480,(0003,1)),DISP=(NEW,DELETE)
//GO.FT21F001 DD UNIT=SYSDA,SPACE=(0024,(0060,1)),DISP=(NEW,DELETE)
//GO.FT14F001 DD UNIT=SYSDA,SPACE=(0032,(0003,1)),DISP=(NEW,DELETE)
//GO.FT15F001 DD UNIT=SYSDA,SPACE=(0016,(0006,1)),DISP=(NEW,DELETE)
//GO.FT17F001 DD UNIT=SYSDA,SPACE=(0032,(0006,1)),DISP=(NEW,DELETE)
//GO.FT19F001 DD UNIT=SYSDA,SPACE=(0480,(0004,1)),DISP=(NEW,DELETE)



          ****** DEFINE FILE STATEMENTS ******


      DEFINE FILE 7(  24,1800,U,I7),8(  16,1800,U,I8),                  DSK00150
     1           10(  56,   6,U,I10),                                   DSK00160
     2           12(  56,  14,U,I12),13(  56,  14,U,I13),               DSK00170
     3           20(   3, 120,U,I20),21(  60,   6,U,I21)                DSK00180



            ****** GENERAL FILE DATA ******


NRX  = NUMBER OF RECORDS OF FILE X
LRBX = LENGTH OF RECORD OF FILE X IN BYTES
LRWX =   -||-       -||-      -||-      WORDS

NR7  =   24     NR8  =   16     NR9  =    4     NR10 =   56     NR12 =   56
LRB7 =7200      LRB8 =7200      LRB9 =   80     LRB10=   24     LRB12=   56
LRW7 =1800      LRW8 =1800                      LRW10=    6     LRW12 =  14
NREC7=   4

NR13 =   56     NR14 =    3     NR15 =    6     NR16 =    0     NR17 =    6
LRB13=   56     LRB14=   32     LRB15=   16     LRB16=    0     LRB17=   32
LRW13=   14

NR19 =    4     NR20 =    3     NR21 =   60     NR22 =    0
LRB19= 480      LRB20= 480      LRB21=   24     LRB22=    0
                LRW20= 120      LRW21=    6

NOTE : IF DATA OF FILES NO 16 OR 22 ARE ZERO THESE FILES ARE NOT REQUIRED
```

****** INPUT DATA ******

```
NEL....TOTAL NUMBER OF ELEMENTS.....  8
NDPT...TOTAL NUMBER OF NODAL POINTS..... 172
NMAT...NUMBER OF DIFFERENT MATERIALS.....  1
       (MAXIMUM IS  10)
NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER.....  96
NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND.....  8
NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL.....  0
NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD.....  4
NN.....NUMBER OF BLOCKS PER COLUMN.....  6
MM.....NUMBER OF BLOCKS PER ROW.....  2
NPEL...NUMBER OF NODES PER ELEMENT.....  32
NREC7..NUMBER OF RECORDS OF FILE NO 7.....  12
```

****** JCL CARDS ******

```
//GO.FT07F001 DD UNIT=SYSDA,SPACE=(6144,(0144,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(6144,(0096,1)),DISP=(NEW,DELETE)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(0128,(0008,1)),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(0024,(0172,1)),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=SYSDA,SPACE=(0056,(0172,1)),DISP=(NEW,DELETE)
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(0056,(0172,1)),DISP=(NEW,DELETE)
//GO.FT20F001 DD UNIT=SYSDA,SPACE=(0768,(0006,1)),DISP=(NEW,DELETE)
//GO.FT21F001 DD UNIT=SYSDA,SPACE=(0024,(0192,1)),DISP=(NEW,DELETE)
//GO.FT14F001 DD UNIT=SYSDA,SPACE=(0032,(0004,1)),DISP=(NEW,DELETE)
//GO.FT15F001 DD UNIT=SYSDA,SPACE=(0016,(0008,1)),DISP=(NEW,DELETE)
//GO.FT17F001 DD UNIT=SYSDA,SPACE=(0032,(0008,1)),DISP=(NEW,DELETE)
//GO.FT19F001 DD UNIT=SYSDA,SPACE=(0480,(0008,1)),DISP=(NEW,DELETE)
```

****** DEFINE FILE STATEMENTS ******

```
 DEFINE FILE 7( 144,1536,U,I7),8(  96,1536,U,I8),                    DSK00190
1            10( 172,   6,U,I10),                                    DSK00200
2            12( 172,  14,U,I12),13( 172,  14,U,I13),                DSK00210
3            20(   6, 192,U,I20),21( 192,   6,U,I21)                 DSK00220
```

****** GENERAL FILE DATA ******

```
NRX  = NUMBER OF RECORDS OF FILE X
LRBX = LENGTH OF RECORD OF FILE X IN BYTES
LRWX =  -II-      -II-     -II-      WORDS

NR7  = 144     NR8  =  96     NR9  =   8     NR10 = 172     NR12 = 172
LRB7 =6144     LRB8 =6144     LRB9 = 128     LRB10=  24     LRB12=  56
LRW7 =1536     LRW8 =1536                    LRW10=   6     LRW12 =  14
NREC7=  12

NR13 = 172     NR14 =   4     NR15 =   8     NR16 =   0     NR17 =   8
LRB13=  56     LRB14=  32     LRB15=  16     LRB16=   0     LRB17=  32
LRW13=  14

NR19 =   8     NR20 =   6     NR21 = 192     NR22 =   0
LRB19= 480     LRB20= 768     LRB21=  24     LRB22=   0
               LRW20= 192     LRW21=   6
```

NOTE : IF DATA OF FILES NO 16 OR 22 ARE ZERO THESE FILES ARE NOT REQUIRED

APPENDIX F

DECK STRUCTURE FOR COMPUTER PROGRAMS IN IBM 360 OS

A. TRISOP - QUADRATIC ELEMENTS

```
//  JOB CARD
//  EXEC FORTHCLG,REGION.FORT=188K,REGION.LINK=118K,REGION.GO=220K
//SYSPRINT DD SYSOUT=A,SPACE=(CYL,(3,1))
//FORT.SYSIN DD *
****  ****
FORTRAN DECK
****  ****
/*
//GO.FT07F001 DD UNIT=SYSDA,SPACE=(LRB7 ,(VR27 ,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(LRB8 ,(VR8 ,1)),DISP=(NEW,DELETE)
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(00080,(VR9 ,1)),DISP=(NEW,DELETE)
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(00024,(VR10,1)),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(00056,(VR12,1)),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=SYSDA,SPACE=(00056,(VR13,1)),DISP=(NEW,DELETE)
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(00032,(VR14,1)),DISP=(NEW,DELETE)
//GO.FT14F001 DD UNIT=SYSDA,SPACE=(00016,(VR15,1)),DISP=(NEW,DELETE)
//GO.FT15F001 DD UNIT=SYSDA,SPACE=(00004,(VR16,1)),DISP=(NEW,DELETE)
//GO.FT16F001 DD UNIT=SYSDA,SPACE=(00032,(VR17,1)),DISP=(NEW,DELETE)
//GO.FT17F001 DD UNIT=SYSDA,SPACE=(00480,(VR19,1)),DISP=(NEW,DELETE)
//GO.FT19F001 DD UNIT=SYSDA,SPACE=(00480,(VR20,1)),DISP=(NEW,DELETE)
//GO.FT20F001 DD UNIT=SYSDA,SPACE=(00024,(VR21,1)),DISP=(NEW,DELETE)
//GO.FT21F001 DD UNIT=SYSDA,SPACE=(00024,(VR22,1)),DISP=(NEW,DELETE)
//GO.FT22F001 DD UNIT=SYSDA,SPACE=(00024,(VR23,1)),DISP=(NEW,DELETE)
****  ****
DATA CARDS
****  ****
```

```
//          JOB CARD
// EXEC FORTHCLG,REGION.FORT=188K,REGION.LINK=118K,REGION.GO=347K
//SYSPRINT DD  SYSOUT=A,SPACE=(CYL,(3,1))
//FORT.SYSIN DD *
                    *                    *
                    **                   **
                    * *                  * *
                    *  *                 *  *
                    *   *                *   *
                    *    *               *    *
                    *   *                *   *
                    *  *                 *  *
                    * *                  * *
                    **                   **
                    *                    *
                       FORTRAN DECK
/
//GO.FT07F001  DD UNIT=SYSDA,SPACE=(LRB7 ,(NR7 ,1)),DISP=(NEW,DELETE)
//GO.FT08F001  DD UNIT=SYSDA,SPACE=(LR88 ,(NR8 ,1)),DISP=(NEW,DELETE)
//GO.FT09F001  DD UNIT=SYSDA,SPACE=(00128,(NR9 ,1)),DISP=(NEW,DELETE)
//GO.FT10F001  DD UNIT=SYSDA,SPACE=(00024,(NR10,1)),DISP=(NEW,DELETE)
//GO.FT12F001  DD UNIT=SYSDA,SPACE=(00056,(NR12,1)),DISP=(NEW,DELETE)
//GO.FT13F001  DD UNIT=SYSDA,SPACE=(00056,(NR13,1)),DISP=(NEW,DELETE)
//GO.FT14F001  DD UNIT=SYSDA,SPACE=(00032,(NR14,1)),DISP=(NEW,DELETE)
//GO.FT15F001  DD UNIT=SYSDA,SPACE=(00016,(NR15,1)),DISP=(NEW,DELETE)
//GO.FT16F001  DD UNIT=SYSDA,SPACE=(00004,(NR16,1)),DISP=(NEW,DELETE)
//GO.FT17F001  DD UNIT=SYSDA,SPACE=(00032,(NR17,1)),DISP=(NEW,DELETE)
//GO.FT19F001  DD UNIT=SYSDA,SPACE=(00480,(NR19,1)),DISP=(NEW,DELETE)
//GO.FT20F001  DD UNIT=SYSDA,SPACE=(00768,(NR20,1)),DISP=(NEW,DELETE)
//GO.FT21F001  DD UNIT=SYSDA,SPACE=(00024,(NR21,1)),DISP=(NEW,DELETE)
//GO.FT22F001  DD UNIT=SYSDA,SPACE=(00024,(NR22,1)),DISP=(NEW,DELETE)
//GO.SYSIN DD *
                    *                    *
                    **                   **
                    * *                  * *
                    *  *                 *  *
                    *   *                *   *
                    *    *               *    *
                    *   *                *   *
                    *  *                 *  *
                    * *                  * *
                    **                   **
                    *                    *
                       DATA CARDS
```

C. AUXILIARY PROGRAM JCL

```
//        JOB CARD
// EXEC FRTCALCP
//FORT.SYSIN DD *
***********************************
*  MAIN PROGRAM - JCL             *
*  (FORTRAN)                      *
***********************************
/*
//ASM.SYSIN DD *
***********************************
*  SUBPROGRAM EXDIC               *
*  (ASSEMBLY LANGUAGE)            *
***********************************
/
//GO.FT08F001 DD UNIT=SYSDA,SPACE=(4,(5,1)),DISP=(NEW,DELETE)
//GO.SYSIN DD *
*****************
*  DATA CARDS   *
*****************
```

APPENDIX G

COMPUTER LISTINGS FOR TRISOP

In this Appendix two computer listings for TRISOP are presented, one for quadratic elements and one for cubic elements.

The two listings are basically the same. The minor existing differences are:

1. Core Space Requirements

The core space requirements for cubic elements are higher than those for quadratic. Those requirements are reflected in the dimensioning of arrays appearing in COMMON and DIMENSION statements.

2. Input of Connectivity Data

The requirement for two headings and also reading of connectivity matrix in two parts for cubic elements dictates an alteration of the corresponding part of subroutine INPUT.

3. Subroutine DISK

Entries RDISK1 and WDISK1 differ in quadratic and cubic elements because optimization of transfer time for element stiffness matrices requires a different partitioning of these matrices into data file records for each case. Because of the similitude of the two programs the listing of TRISOP for cubic elements is not presented complete. Only the parts that differ from the listing for quadratic elements are shown.

Listing for subroutine CUB is not given in this Appendix since there is a choice of different subroutines each corresponding to integration by use of a different number of integration points. Computer listings for these subroutines are presented in Appendix I.

```
C                                                                                MAI00010
C                                                                                MAI00015
C                                                                                MAI00020
CCCCC            CONTROL PROGRAM                                                 MAI00025
C                                                                                MAI00030
CCCCC                                                                            MAI00035
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,           MAI00040
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                      MAI00050
      COMMON/NB3/NR7,LRB7,LRW7,NNRC,NR8,LRB8,LRW8,NR9,LRB9,NR10,LRB10,NR16       MAI00060
     1LRW10,NR12,LRB12,LRW12,NR13,LRB13,LRW13,NR14,LRB14,NR15,LRB15,             MAI00070
     2,LRB16,NR17,LRB17,NR19,LRB19,NR20,LRB20,LRW20,NR21,LRB21,LRW21,            MAI00080
     3NR22,LRB22                                                                 MAI00090
      DIMENSION NFR(36)                                                          MAI00100
      EQUIVALENCE (NFR(1),NR7)                                                   MAI00110
C                                                                                MAI00120
      READ(5,1000) NFR                                                           MAI00130
      NREC7=NNRC                                                                 MAI00140
C                                                                                MAI00150
CCC           INITIALIZATION OF THE PROGRAM                                      MAI00160
C                                                                                MAI00170
      CALL MAINE                                                                 MAI00180
C                                                                                MAI00190
 1000 FORMAT(7I10)                                                               MAI00200
      STOP                                                                       MAI00210
      END
C
CC*****************************************************************************
CC*****************************************************************************
C
      SUBROUTINE MAINE                                                           SMA00020
C                                                                                SMA00030
      IMPLICIT REAL*8(A-H,O-Z)                                                   SMA00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,           SMA00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                      SMA00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),              SMA00070
     1SSNN1(7)                                                                   SMA00080
      DIMENSION SSSS(6),SSNN(6)                                                  SMA00090
      EQUIVALENCE (SSSS(1),SSSS1(1)),(SSNN(1),SSNN1(1))                          SMA00100
C                                                                                SMA00110
CC*****************************************************************************SMA00120
CC                                                                               SMA00130
CCC            INPUT                                                             SMA00140
C                                                                                SMA00150
   10 CALL SETIME                                                                SMA00160
      CLOCK=ITIME(0)*0.01                                                        SMA00170
      READ(5,950) CHKWD                                                          SMA00180
```

```
      ENTRY MAINE1                                                      SMA00190
      CALL INPUT                                                        SMA00200
      CALL GETIME(IET)                                                  SMA00210
      CPUTM=IET*0.000026                                                SMA00220
      WRITE(6,1000)                                                     SMA00230
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA00240
      WRITE(6,8000) CLOCK,CPUTM                                         SMA00250
C*****************************************************************     SMA00260
CCCCC                                                                   SMA00270
CCCCC          FORMATION OF ELEMENT NODAL COORDINATE ARRAYS             SMA00280
CCCCC                                                                   SMA00290
C                                                                       SMA00300
      CALL SETIME                                                       SMA00310
      CLOCK=ITIME(0)*0.01                                               SMA00320
      CALL SORT                                                         SMA00330
      CALL GETIME(IET)                                                  SMA00340
      CPUTM=IET*0.000026                                                SMA00350
      WRITE(6,2000)                                                     SMA00360
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA00370
      WRITE(6,8000) CLOCK,CPUTM                                         SMA00380
C*****************************************************************     SMA00390
CCCCC                                                                   SMA00400
CCCCC          FORMATION OF ELEMENT STIFFNESS MATRICES                  SMA00410
CCCCC                                                                   SMA00420
C                                                                       SMA00430
      CALL SETIME                                                       SMA00440
      CLOCK=ITIME(0)*0.01                                               SMA00450
      CALL FSTF                                                         SMA00460
      CALL GETIME(IET)                                                  SMA00470
      CPUTM=IET*0.000026                                                SMA00480
      WRITE(6,3000)                                                     SMA00490
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA00500
      WRITE(6,8000) CLOCK,CPUTM                                         SMA00510
C*****************************************************************     SMA00520
CCCCC                                                                   SMA00530
CCCCC          FORMATION OF TOTAL STIFFNESS MATRIX                      SMA00540
CCCCC                                                                   SMA00550
C                                                                       SMA00560
      CALL SETIME                                                       SMA00570
      CLOCK=ITIME(0)*0.01                                               SMA00580
      CALL MERGE                                                        SMA00590
      CALL GETIME(IET)                                                  SMA00600
      CPUTM=IET*0.000026                                                SMA00610
      WRITE(6,5000)                                                     SMA00620
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA00630
      WRITE(6,8000) CLOCK,CPUTM                                         SMA00640
C*****************************************************************     SMA00650
C                                                                       SMA00660
```

```
C                                                                       SMA00670
C                                                                       SMA00680
C             FORMATION OF LOAD VECTOR                                  SMA00690
C                                                                       SMA00700
      CALL SETIME                                                       SMA00710
      CLOCK=ITIME(0)*0.01                                               SMA00720
      CALL FLOAD                                                        SMA00730
      CALL GETIME(IET)                                                  SMA00740
      CPUTM=IET*0.000026                                                SMA00750
      WRITE(6,6000)                                                     SMA00760
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA00770
      WRITE(6,8000) CLOCK,CPUTM                                         SMA00780
C*****************************************************************     SMA00790
C*****************************************************************     SMA00800
C                                                                       SMA00810
C             APPLICATION OF BOUNDARY CONDITIONS                        SMA00820
C                                                                       SMA00830
C                                                                       SMA00840
C             SELECT THE APPROPRIATE SEGMENT (1) OR (2) DEPENDING ON    SMA00850
C             WHETHER THE B.C.'S INCLUDE PREDEFINED DISPLACEMENTS OR    SMA00860
C             NOT                                                       SMA00870
C                                                                       SMA00880
      IF(NPBC2.NE.0) GO TO 100                                          SMA00890
C                                                                       SMA00900
C                 (1) THIS SEGMENT REFERS TO B.C.'S WITH NO PREDEFINED  SMA00910
C                     DISPLACEMENTS                                     SMA00920
C                                                                       SMA00930
      CALL SETIME                                                       SMA00940
      CLOCK=ITIME(0)*0.01                                               SMA00950
      CALL BCOND                                                        SMA00960
      CALL GETIME(IET)                                                  SMA00970
      CPUTM=IET*0.000026                                                SMA00980
      WRITE(6,7000)                                                     SMA00990
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA01000
      WRITE(6,8000) CLOCK,CPUTM                                         SMA01010
      GO TO 200                                                         SMA01020
C                                                                       SMA01030
C                 (2) THIS SEGMENT REFERS TO B.C.'S INCLUDING PREDEFINED SMA01040
C                     DISPLACEMENTS                                     SMA01050
C                                                                       SMA01060
  100 CALL SETIME                                                       SMA01070
      CLOCK=ITIME(0)*0.01                                               SMA01080
      CALL BCOND2                                                       SMA01090
      CALL GETIME(IET)                                                  SMA01100
      CPUTM=IET*0.000026                                                SMA01110
      WRITE(6,7500)                                                     SMA01120
      CLOCK=ITIME(0)*0.01-CLOCK                                         SMA01130
      WRITE(6,8000) CLOCK,CPUTM                                         SMA01140
C
```

```
C*****************************************************************************SMA01150
C                                                                               SMA01160
CCC             SOLUTION OF THE SYSTEM OF EQUATIONS                             SMA01170
C                                                                               SMA01180
  200 CALL SETIME                                                               SMA01190
      CLOCK=ITIME(0)*0.01                                                       SMA01200
      CALL SOLVE                                                                SMA01210
      CALL GETIME(IET)                                                          SMA01220
      CPUTM=IET*0.000026                                                        SMA01230
      WRITE(6,9000)                                                             SMA01240
      CLOCK=ITIME(0)*0.01-CLOCK                                                 SMA01250
      WRITE(6,8000) CLOCK,CPUTM                                                 SMA01260
C                                                                               SMA01270
C*****************************************************************************SMA01280
C                                                                               SMA01290
CCC             DETERMINATION OF NODAL DISPLACEMENTS AND REACTIONS AT           SMA01300
CCC             THE CONSTRAINED POINTS                                          SMA01310
C                                                                               SMA01320
      CALL SETIME                                                               SMA01330
      CLOCK=ITIME(0)*0.01                                                       SMA01340
      CALL DISP                                                                 SMA01350
      CALL GETIME(IET)                                                          SMA01360
      CPUTM=IET*0.000026                                                        SMA01370
      WRITE(6,9100)                                                             SMA01380
      CLOCK=ITIME(0)*0.01-CLOCK                                                 SMA01390
      WRITE(6,8000) CLOCK,CPUTM                                                 SMA01400
C                                                                               SMA01410
C*****************************************************************************SMA01420
C                                                                               SMA01430
CCC             COMPUTATION OF ELEMENT NODAL STRESSES AND STRAINS               SMA01440
C                                                                               SMA01450
      CALL SETIME                                                               SMA01460
      CLOCK=ITIME(0)*0.01                                                       SMA01470
      REWIND 9                                                                  SMA01480
      REWIND 19                                                                 SMA01490
      DO 400 I=1,7                                                              SMA01500
      SSNN1(I)=0.0D0                                                            SMA01510
  400 SSSS1(I)=0.0D0                                                            SMA01520
      DO 410 I=1,NDPT                                                           SMA01530
      CALL WDISKS(I)                                                            SMA01540
  410 CALL WDISKN(I)                                                            SMA01550
C                                                                               SMA01560
CCC             SELECT THE APPROPRIATE SEGMENT (1) OR (2) DEPENDING ON          SMA01570
CCC             WHETHER ONE OR MORE MATERIALS ARE PRESENT IN THE STRUCTURE      SMA01580
C                                                                               SMA01590
      IF(NMAT.EQ.1) GO TO 550                                                   SMA01600
C                                                                               SMA01610
CC              (1) THIS SEGMENT IS USED FOR THE CASE OF MORE THEN ONE          SMA01620
```

```
C                    MATERIALS PRESENT IN THE STRUCTURE                      SMA01630
C                                                                            SMA01640
C                                                                            SMA01650
      NMAT1=0                                                                SMA01660
      REWIND 16                                                              SMA01670
      DO 500 I=1,NEL                                                         SMA01680
      READ(19) COREL                                                         SMA01690
      READ(16) NCON21                                                        SMA01700
      IF(NMAT1.EQ.NCON21) GO TO 450                                          SMA01710
      NMAT1=NCON21                                                           SMA01720
      CALL ELPROP(NMAT1)                                                     SMA01730
  450 CALL STRESS(I)                                                         SMA01740
  500 CONTINUE                                                               SMA01750
      GO TO 650                                                              SMA01760
C                                                                            SMA01770
CC                (2) THIS SEGMENT IS USED FOR THE CASE OF ONE MATERIAL      SMA01780
C                                                                            SMA01790
  550 DO 600 I=1,NEL                                                         SMA01800
      READ(19) COREL                                                         SMA01810
  600 CALL STRESS(I)                                                         SMA01820
C*************************************************************************SMA01830
C                                                                            SMA01840
C                COMPUTATION OF AVERAGE NODAL STRESSES AND STRAINS           SMA01850
C                                                                            SMA01860
C                                                                            SMA01870
C                                                                            SMA01880
C                  (1) AVERAGE STRESSES                                      SMA01890
C                                                                            SMA01900
  650 NPAGE=NDPT/NLINE                                                       SMA01910
      NL=NPAGE*NLINE                                                         SMA01920
      NL1=NL+1                                                               SMA01930
      NLAST=NDPT-NL                                                          SMA01940
      IF(NPAGE.EQ.0) GO TO 710                                               SMA01950
      DO 705 I=1,NPAGE                                                       SMA01960
      WRITE(6,9700)                                                          SMA01970
      I1=(I-1)*NLINE+1                                                       SMA01980
      I2=I1+NLINE-1                                                          SMA01990
      DO 705 J=I1,I2                                                         SMA02000
      CALL RDISKS(J)                                                         SMA02010
      DO 700 K=1,6                                                           SMA02020
  700 SSSS1(K)=SSSS1(K)/SSSS1(7)                                             SMA02030
  705 WRITE(6,9600) J,SSSS                                                   SMA02040
      IF(NLAST.EQ.0) GO TO 730                                               SMA02050
  710 WRITE(6,9700)                                                          SMA02060
      DO 725 I=NL1,NDPT                                                      SMA02070
      CALL RDISKS(I)                                                         SMA02080
      DO 720 K=1,6                                                           SMA02090
  720 SSSS1(K)=SSSS1(K)/SSSS1(7)                                             SMA02100
  725 WRITE(6,9600) I,SSSS
```

```
  730 CONTINUE                                                                  SMA02110
C                                                                               SMA02120
CC                (2) AVERAGE STRAINS                                           SMA02130
C                                                                               SMA02140
      IF(NPAGE.EQ.0) GO TO 745                                                  SMA02150
      DO 740 I=1,NPAGE                                                          SMA02160
      WRITE(6,9500)                                                             SMA02170
      I1=(I-1)*NLINE+1                                                          SMA02180
      I2=I1+NLINE-1                                                             SMA02190
      DO 740 J=I1,I2                                                            SMA02200
      CALL RDISKN(J)                                                            SMA02210
      DO 735 K=1,6                                                              SMA02220
  735 SSNN1(K)=SSNN1(K)/SSNN1(7)                                                SMA02230
  740 WRITE(6,9600) J,SSNN                                                      SMA02240
      IF(NLAST.EQ.0) GO TO 760                                                  SMA02250
  745 WRITE(6,9500)                                                             SMA02260
      DO 755 I=NL1,NDPT                                                         SMA02270
      CALL RDISKN(I)                                                            SMA02280
      DO 750 K=1,6                                                              SMA02290
  750 SSNN1(K)=SSNN1(K)/SSNN1(7)                                                SMA02300
  755 WRITE(6,9600) I,SSNN                                                      SMA02310
  760 CONTINUE                                                                  SMA02320
      CALL GETIME(IET)                                                          SMA02330
      CPUTM=IET*0.000026                                                        SMA02340
      WRITE(6,9200)                                                             SMA02350
      CLOCK=ITIMF(0)*0.01-CLOCK                                                 SMA02360
      WRITE(6,8000) CLOCK,CPUTM                                                 SMA02370
C                                                                               SMA02380
C*******************************************************************************SMA02390
C                                                                               SMA02400
      GO TO 10                                                                  SMA02410
C                                                                               SMA02420
  950 FORMAT(1A8)                                                               SMA02430
 1000 FORMAT(5X,'***** INPUT *****')                                            SMA02440
 2000 FORMAT(5X,'***** SORT *****')                                             SMA02450
 3000 FORMAT(5X,'***** FSTF *****')                                             SMA02460
 4000 FORMAT(5X,'***** CUB *****')                                              SMA02470
 5000 FORMAT(5X,'***** MERGE *****')                                            SMA02480
 6000 FORMAT(5X,'***** FLOAD *****')                                            SMA02490
 7000 FORMAT(5X,'***** BCOND *****')                                            SMA02500
 7500 FORMAT(5X,'***** BCOND2 *****')                                           SMA02510
 9000 FORMAT(5X,'***** SOLVE *****')                                            SMA02520
 9100 FORMAT(5X,'***** DISP *****')                                             SMA02530
 9200 FORMAT(5X,'***** STRESS *****')                                           SMA02540
 8000 FORMAT(5X,7H TIME =,1F7.2,8H SECONDS,6H(CPU =,1F7.2,9H SECONDS)//)        SMA02560
 9500 FORMAT(1H1,//,' A-V-E-R-A-G-E  S-T-R-A-I-N-S  AT  THE  JOINTS ',         SMA02570
     1//,2X,'JOINT',                                                            SMA02580
     29X,'SNX',12X,'SNY',12X,'SNZ',12X,'SNXY',11X,'SNYZ',11X,'SNZX',//)         SMA02590
```

```
 9600 FORMAT(2X,I4,4X,6(1PD15.6))                                        SMA02600
 9700 FORMAT(1H1,//,' A-V-E-R-A-G-E  S-T-R-E-S-S-E-S  AT  THE  JOINTS ',SMA02610
     1//,2X,'JOINT',                                                     SMA02620
     29X,'SSX',12X,'SSY',12X,'SSZ',12X,'SSXY',11X,'SSYZ',11X,'SSZX',//) SMA02630
      END                                                                SMA02640
C
C
C*****************************************************************************
C*****************************************************************************
C


      SUBROUTINE INPUT                                                   INP00020
C                                                                        INP00030
      IMPLICIT REAL*8(A-H,O-Z)                                           INP00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,NM,NS,NCOUNT,NST,NSTF,NCLD,   INP00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL              INP00060
      COMMON /NB2/ NCON(20),NBC(4)                                       INP00070
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                           INP00080
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),      INP00090
     1SSNN1(7)                                                           INP00100
      DIMENSION TITLE(10),FMT(10),CLOAD1(3)                              INP00110
      EQUIVALENCE (CLOAD(1),CLOAD1(1)),(TITLE(1),COREL(1,1)),            INP00120
C                                                                        INP00140
     1(FMT(1),COREL(1,2))                                                INP00130
      PI=3.14159265359                                                   INP00150
      REWIND 9                                                           INP00160
      REWIND 14                                                          INP00170
      REWIND 15                                                          INP00180
      REWIND 16                                                          INP00190
C                                                                        INP00200
C*****************************************************************************INP00210
C                                                                        INP00220
C              TITLE AND GENERAL CHARACTERISTICS OF THE PROBLEM          INP00230
C                                                                        INP00240
      READ(5,1000) TITLE                                                 INP00250
      WRITE(6,1000) TITLE                                                INP00260
      READ(5,2000) NEL,NDPT,NS,NPEL,NMAT,NCLD,NPBC,NPBC2,NLINE,KEL       INP00270
C                                                                        INP00280
C              STOP TRAP FOR LAST PROBLEM                                INP00290
C                                                                        INP00300
      IF(NEL.LT.0) STOP                                                  INP00310
C                                                                        INP00320
      WRITE(6,3000) NEL,NDPT,NMAT,NS,NPBC,NPBC2,NCLD                     INP00330
      NDF=3                                                              INP00340
      NEQ=NDPT*NDF                                                       INP00350
      NCOUNT=NS*NS                                                       INP00360
      NST=NPEL*NDF                                                       INP00370
```

```
      NSTF=NST*NST                                                      INP00380
      NSB=6*NST                                                         INP00390
      NN=(NEQ+NS-1)/NS                                                  INP00400
C                                                                       INP00410
C*****************************************************************    INP00420
C                                                                       INP00430
C             STOP TRAPS FOR WRONG CHARACTERISTICS OF FILES (EXCEPT NR7) INP00440
C             STOP TRAP FOR WRONG NUMBER OF MATERIALS                   INP00450
C                                                                       INP00460
      CALL ERROR1(NMAT)                                                 INP00470
C*****************************************************************    INP00490
C                                                                       INP00480
C                                                                       INP00500
C             CONNECTIVITY MATRIX                                       INP00510
C                                                                       INP00520
      READ(5,1000) TITLE                                                INP00530
      READ(5,1000) FMT                                                  INP00540
      WRITE(6,1100)                                                     INP00550
      WRITE(6,1000) TITLE                                               INP00560
      WRITE(6,1100)                                                     INP00570
      NBAND=0                                                           INP00580
C                                                                       INP00590
C             SELECT THE APPROPRIATE SEGMENT (1) OR (2) DEPENDING ON    INP00600
C             WHETHER ONE OR MORE MATERIALS ARE PRESENT IN THE STRUCTURE INP00610
C                                                                       INP00620
      IF(NMAT.EQ.1) GO TO 150                                           INP00630
C                                                                       INP00640
C               (1) THIS SEGMENT IS USED FOR THE CASE OF MORE THEN ONE  INP00650
C                   MATERIALS PRESENT IN THE STRUCTURE                  INP00660
      DO 100 I=1,NEL                                                    INP00670
      READ(5,FMT) NCON,NCON21                                           INP00680
      WRITE(6,FMT) NCON,NCON21                                          INP00690
      WRITE(9) NCON                                                     INP00700
      WRITE(16) NCON21                                                  INP00710
C                                                                       INP00720
C             COMPUTATION OF HALF BANDWIDTH                             INP00730
C                                                                       INP00740
      NPELM=NPEL-1                                                      INP00750
      DO 100 J=1,NPELM                                                  INP00760
      JP=J+1                                                            INP00770
      DO 100 K=JP,NPEL                                                  INP00780
  100 NBAND=MAXO(NBAND,IABS(NCON(J)-NCON(K)))                           INP00790
      GO TO 200                                                         INP00800
C                                                                       INP00810
C                                                                       INP00820
C               (2) THIS SEGMENT IS USED FOR THE CASE OF ONE MATERIAL   INP00830
C                                                                       INP00840
  150 DO 160 I=1,NEL                                                    INP00850
```

```
      READ(5,FMT) NCON                                                  INP00860
      WRITE(6,FMT) NCON                                                 INP00870
      WRITE(9) NCON                                                     INP00880
C                                                                       INP00890
CC           COMPUTATION OF HALF BANDWIDTH                              INP00900
C                                                                       INP00910
      NPELM=NPEL-1                                                      INP00920
      DO 160 J=1,NPELM                                                  INP00930
      JP=J+1                                                            INP00940
      DO 160 K=JP,NPEL                                                  INP00950
  160 NBAND=MAX0(NBAND,IABS(NCON(J)-NCON(K)))                           INP00960
C                                                                       INP00970
  200 NBAND=(NBAND+1)*NDF                                               INP00980
      MM=(NBAND+2*(NS-1))/NS                                            INP00990
      IF(NN.LT.MM) MM=NN                                                INP01000
       WRITE(6,4000) NEQ,NBAND,NN,MM,NCOUNT,NSTF                        INP01010
C                                                                       INP01020
C*************************************************************************INP01030
C                                                                       INP01040
CC           STOP TRAP FOR WRONG NUMBER OF RECORDS OF FILE NO 7         INP01050
C                                                                       INP01060
      CALL ERROR2                                                       INP01070
C                                                                       INP01080
C*************************************************************************INP01090
C                                                                       INP01100
CC           INITIALIZATION OF VARIABLES OF SUBROUTINE 'DISK'           INP01110
C                                                                       INP01120
      CALL DISK                                                         INP01130
C                                                                       INP01140
C*************************************************************************INP01150
C                                                                       INP01160
CC          NODAL POINT COORDINATES                                     INP01170
C                                                                       INP01180
      WRITE(6,1100)                                                     INP01190
      READ(5,1000) TITLE                                                INP01200
      WRITE(6,1000) TITLE                                               INP01210
      WRITE(6,1100)                                                     INP01220
      READ(5,1000) FMT                                                  INP01230
      DO 300 I=1,NDPT                                                   INP01240
      READ(5,FMT) IEL,COORD,KND                                         INP01250
C                                                                       INP01260
CC           IF GIVEN COORDINATES ARE CARTESIAN SKIP NEXT SEGMENT       INP01270
C                                                                       INP01280
      IF(KND.EQ.0) GO TO 250                                            INP01290
C                                                                       INP01300
CC           CONVERSION OF CYLINDRICAL COORDINATES TO CARTESIAN         INP01310
C                                                                       INP01320
      PHI=PI*COORD(2)/180.0D0                                           INP01330
```

```
      X=COORD(1)*DCOS(PHI)                                                  INP01340
      Y=COORD(1)*DSIN(PHI)                                                  INP01350
      COORD(1)=X                                                            INP01360
      COORD(2)=Y                                                            INP01370
  250 WRITE(6,FMT) IEL,COORD,KND                                            INP01380
  300 CALL WDISKC(I)                                                        INP01390
      WRITE(6,1100)                                                         INP01400
C                                                                           INP01410
CC*************************************************************************INP01420
C                                                                           INP01430
CC          MATERIAL PROPERTIES                                             INP01440
C                                                                           INP01450
      READ(5,1000) TITLE                                                    INP01460
      WRITE(6,1000)TITLE                                                    INP01470
      WRITE(6,1100)                                                         INP01480
      READ(5,1000) FMT                                                      INP01490
      DO 400 I=1,NMAT                                                       INP01500
      READ(5,FMT) N,(ELDAT(N,J),J=1,3)                                      INP01510
  400 WRITE(6,FMT) N,(ELDAT(N,J),J=1,3)                                     INP01520
      WRITE(6,1100)                                                         INP01530
C                                                                           INP01540
CC*************************************************************************INP01550
CC                                                                          INP01560
CC          CONCENTRATED LOAD DATA                                          INP01570
C                                                                           INP01580
      READ(5,1000) TITLE                                                    INP01590
      WRITE(6,1000)TITLE                                                    INP01600
      WRITE(6,1100)                                                         INP01610
      NDFP=NDF+1                                                            INP01620
      READ(5,1000) FMT                                                      INP01630
      DO 500 I=1,NCLD                                                       INP01640
      READ(5,FMT)    NCL,CLOAD1                                             INP01650
      WRITE(6,FMT)     NCL,CLOAD1                                           INP01660
      ANCL=NCL                                                              INP01670
      CLOAD(NDFP)=ANCL                                                      INP01680
  500 WRITE(14) CLOAD                                                       INP01690
C                                                                           INP01700
CC*************************************************************************INP01710
CC                                                                          INP01720
CC          BOUNDARY CONDITIONS                                             INP01730
CCC                                                                         INP01740
CCC                                                                         INP01750
CCC             (1) BOUNDARY CONDITION CODES                                INP01760
C                                                                           INP01770
      READ(5,1000) TITLE                                                    INP01780
      WRITE(6,1000) TITLE                                                   INP01790
      WRITE(6,1100)                                                         INP01800
      READ(5,1000) FMT                                                      INP01810
```

```
      READ(5,FMT) NBC                                                    INP01820
      WRITE(6,FMT) NBC                                                   INP01830
      NBCL=NBC(1)                                                        INP01840
      NBC(1)=NDF*NBC(1)-NDF                                              INP01850
      WRITE(15) NBC                                                      INP01860
      NPBCL=NPBC-1                                                       INP01870
      DO 600 I=2,NPBCL                                                   INP01880
      READ(5,FMT) NBC                                                    INP01890
      WRITE(6,FMT) NBC                                                   INP01900
      NBC(1)=NDF*NBC(1)-NDF                                              INP01910
  600 WRITE(15) NBC                                                      INP01920
      READ(5,FMT) NBC                                                    INP01930
      WRITE(6,FMT) NBC                                                   INP01940
      NBCH=NBC(1)                                                        INP01950
      NBC(1)=NDF*NBC(1)-NDF                                              INP01960
      WRITE(15) NBC                                                      INP01970
C                                                                        INP01980
CC                  IF THE B.C.'S DO NOT INCLUDE PREDEFINED DISPLACEMENTSINP01990
CC                  SKIP NEXT SEGMENT                                  INP02000
C                                                                        INP02010
      IF(NPBC2.EQ.0) RETURN                                              INP02020
C                                                                        INP02030
CC                (2) PREDEFINED DISPLACEMENTS                         INP02040
C                                                                        INP02050
      READ(5,1000) TITLE                                                 INP02060
      WRITE(6,1000) TITLE                                                INP02070
      WRITE(6,1100)                                                      INP02080
      READ(5,1000) FMT                                                   INP02090
      DO 650 I=1,NPBC2                                                   INP02100
      READ(5,FMT) NDBC,BCON                                              INP02110
      WRITE(6,FMT) NDBC,BCON                                             INP02120
  650 WRITE(22) BCON                                                     INP02130
C                                                                        INP02140
C*************************************************************************INP02150
C                                                                        INP02160
      RETURN                                                             INP02170
C                                                                        INP02180
 1000 FORMAT(10A8)                                                       INP02190
 1100 FORMAT(///)                                                        INP02200
 2000 FORMAT(10I5)                                                       INP02210
 3000 FORMAT(//,                                                         INP02220
     15X,'NEL....TOTAL NUMBER OF ELEMENTS.....................',I3/,    INP02230
     25X,'NDPT...TOTAL NUMBER OF NODAL POINTS.................',I3/,    INP02240
     35X,'NMAT...NUMBER OF DIFFERENT MATERIALS................',I3/,    INP02250
     45X,'        (MAXIMUM IS  10)',/,                                   INP02260
     55X,'NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER.....',I3/,    INP02270
     65X,'NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND.....',I3/,     INP02280
     75X,'NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL....',I3/,    INP02290
```

```
     85X,'NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD.....',I3/)   INP02300
 4000 FORMAT(//,                                                     INP02310
     15X,'NEQ....TOTAL NUMBER OF EQUATIONS.....',I5/,                INP02320
     25X,'NBAND..HALF-BAND WIDTH OF THE SYSTEM.....',I5/,            INP02330
     35X,'NN.....NUMBER OF BLOCKS PER COLUMN.....',I5/,              INP02340
     45X,'MM.....NUMBER OF BLOCKS PER ROW.....',I5/,                 INP02350
     55X,'NCOUNT.NUMBER OF COEFFICIENTS PER BLOCK.....',I5/,         INP02360
     65X,'NSTF...NUMBER OF COEFFICIENTS PER ELEMENT.....',I5/)       INP02370
C                                                                    INP02380
      END                                                            INP02390
C
C
C*********************************************************************
C*********************************************************************
C


      SUBROUTINE ELPROP(I)                                           ELP00020
      IMPLICIT REAL*8(A-H,O-Z)                                       ELP00030
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                       ELP00040
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(10,3),UJNT(3),SSSS1(7),  ELP00050
     1SSNN1(7)                                                       ELP00060
      DO 200 L=1,6                                                   ELP00070
      DO 200 J=1,6                                                   ELP00080
  200 ELAST(L,J)=0.0D0                                               ELP00090
      E=ELDAT(I,1)                                                   ELP00100
      PR=ELDAT(I,2)                                                  ELP00110
      ER=E/((1.0D0+PR)*(1.0D0-2.0D0*PR))                             ELP00120
      EK=ER*(1.0D0-PR)                                               ELP00130
      EL=ER*PR                                                       ELP00140
      ELAST(1,1)=EK                                                  ELP00150
      ELAST(1,2)=EL                                                  ELP00160
      ELAST(1,3)=EL                                                  ELP00170
      ELAST(2,1)=EL                                                  ELP00180
      ELAST(2,2)=EK                                                  ELP00190
      ELAST(2,3)=EL                                                  ELP00200
      ELAST(3,1)=EL                                                  ELP00210
      ELAST(3,2)=EL                                                  ELP00220
      ELAST(3,3)=EK                                                  ELP00230
      ELAST(4,4)=ER*(1.0D0-2.0D0*PR)/2.0D0                           ELP00240
      ELAST(5,5)=ELAST(4,4)                                          ELP00250
      ELAST(6,6)=ELAST(4,4)                                          ELP00260
      RETURN                                                         ELP00270
      END                                                            ELP00280
C
C
C*********************************************************************
C*********************************************************************
```

```
C
      SUBROUTINE FSTF                                                    FST00020
C                                                                        FST00030
      IMPLICIT REAL*8(A-H,O-Z)                                           FST00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,   FST00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL              FST00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),      FST00070
     1SSNN1(7)                                                           FST00080
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),      FST00090
     1RB3(60)                                                            FST00100
      DIMENSION STK(3600),AK(3600),B(360)                                FST00110
      EQUIVALENCE (AK1(1,1),STK(1)),(AK2(1,1),AK(1)),(AK3(1,1),B(1))     FST00120
C                                                                        FST00130
CCC           SELECT THE APPROPRIATE SEGMENT (1) OR (2) DEPENDING ON     FST00140
CCC           WHETHER ONE OR MORE MATERIALS ARE PRESENT IN THE STRUCTUREFST00150
C                                                                        FST00160
      REWIND 19                                                          FST00170
      IF(NMAT.EQ.1) GO TO 300                                            FST00180
C                                                                        FST00190
CCC             (1) THIS SEGMENT IS USED FOR THE CASE OF MORE THEN ONE   FST00200
CCC                 MATERIALS PRESENT IN THE STRUCTURE                   FST00210
C                                                                        FST00220
      REWIND 16                                                          FST00230
      N2=-1                                                              FST00240
      DO 200 I=1,NEL                                                     FST00250
      READ(19) COREL                                                     FST00260
      READ(16) NCON21                                                    FST00270
      N=NCON21                                                           FST00280
      IF(N.EQ.N2) GO TO 100                                              FST00290
      CALL ELPROP(N)                                                     FST00300
      N2=N                                                               FST00310
  100 CALL CUB                                                           FST00320
      CALL WDISK1(I)                                                     FST00330
  200 CONTINUE                                                           FST00340
      RETURN                                                             FST00350
C                                                                        FST00360
CCC             (2) THIS SEGMENT IS USED FOR THE CASE OF ONE MATERIAL    FST00370
C                                                                        FST00380
  300 CALL ELPROP(1)                                                     FST00390
      DO 400 I=1,NEL                                                     FST00400
      READ(19) COREL                                                     FST00410
      CALL CUB                                                           FST00420
      CALL WDISK1(I)                                                     FST00430
  400 CONTINUE                                                           FST00440
      RETURN                                                             FST00450
      END                                                                FST00460
```

```
C
CCC*****************************************************************************
CCC*****************************************************************************
C
      SUBROUTINE CUB
C
CCCCC
C     LISTINGS OF SUBROUTINES PERFORMING NUMERICAL INTEGRATION BY USE OF
C     TWO, THREE, FOUR, OR FIVE GAUSS POINTS ARE SHOWN IN APPENDIX
CCCCCCCC
CCCCC*****************************************************************************
CCC*****************************************************************************
C
      SUBROUTINE FORMK(X,Y,Z,INDIC)                                          FMK00020
                                                                             FMK00030
C     IMPLICIT REAL*8(A-H,O-Z)                                               FMK00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,       FMK00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,KLINE,NREC7,KEL                  FMK00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),          FMK00070
     1SSNN1(7)                                                               FMK00080
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),          FMK00090
     1RB3(60)                                                                FMK00100
      DIMENSION   STK(60,60),AK(60,60),B(6,60),B1(60,6),W1(3,20),            FMK00110
     1DNX(3,20),AJIN(3,3),AJ(3,3),CORDG(20,3)                                FMK00120
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),            FMK00130
     1AK3(1,1)),(B1(1,1),AK3(1,7)),(W1(1,1),RB2(1))                          FMK00140
      DATA CORDG/1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.0D0,         FMK00150
     11.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,     FMK00160
     21.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,       FMK00170
     31.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,      FMK00180
     4-1.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,         FMK00190
     50.0D0,0.0D0,0.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,    FMK00200
     6-1.0D0,-1.0D0/                                                         FMK00210
C                                                                            FMK00220
CC*****************************************************************************FMK00230
CC                                                                           FMK00240
CC           GENERAL FORMULAE FOR COMPUTATION OF DERIVATIVES OF SHAPE        FMK00250
C            FUNCTIONS WITH RESPECT TO 'XI','ETA','ZETA'                     FMK00260
```

```
C                                                                       FMK00270
CC                (1) CORNER NODES                                      FMK00280
C                                                                       FMK00290
      DF(C,D,E,X1,Y1,Z1)=X1*(1.0D0+D*Y1)*(1.0D0+E*Z1)*(2.0D0*C*X1+D*Y1+ FMK00300
     1E*Z1-1.0D0)/8.0D0                                                 FMK00310
C                                                                       FMK00320
CC                (2) MIDSIDE NODES                                     FMK00330
C                                                                       FMK00340
      D2(C,D,E,    Y1,Z1)=-C*(1.0D0+D*Y1)*(1.0D0+E*Z1)/2.0D0            FMK00350
      D4(C,  E,    Y1,Z1)=(1.0D0-C*C)*Y1*(1.0D0+E*Z1)/4.0D0             FMK00360
C                                                                       FMK00370
C***********************************************************************FMK00380
C                                                                       FMK00390
CC            COMPUTATION OF DERIVATIVES OF SHAPE FUNCTIONS WITH RESPECT FMK00400
CC            TO 'XI','ETA','ZETA' EVALUATED AT THE POINTS OF INTEGRATIONFMK00410
CC                                                                      FMK00420
CC                (1) CORNER NODES :    XI= + OR - 1                    FMK00430
CC                                    ZETA= + OR - 1                    FMK00440
CC                                     ETA= + OR - 1                    FMK00450
CC                                                                      FMK00460
CC                              NODE # : 1,3,5,7,13,15,17,19            FMK00470
C                                                                       FMK00480
      DO 100 I=1,2                                                      FMK00490
      II=12*(I-1)+1                                                     FMK00500
      IT=II+6                                                           FMK00510
      DO 100 J=II,IT,2                                                  FMK00520
      X1=CORDG(J,1)                                                     FMK00530
      Y1=CORDG(J,2)                                                     FMK00540
      Z1=CORDG(J,3)                                                     FMK00550
      W1(1,J)=DF(X,Y,Z,X1,Y1,Z1)                                        FMK00560
      W1(2,J)=DF(Y,Z,X,Y1,Z1,X1)                                        FMK00570
  100 W1(3,J)=DF(Z,X,Y,Z1,X1,Y1)                                        FMK00580
C                                                                       FMK00590
CC                (2) MIDSIDE NODES :   XI= 0                           FMK00600
CC                                    ZETA= + OR - 1                    FMK00610
CC                                     ETA= + OR - 1                    FMK00620
CC                                                                      FMK00630
CC                              NODE # : 2,6,14,18                      FMK00640
C                                                                       FMK00650
      DO 200 K=1,2                                                      FMK00660
      II=(K-1)*12+2                                                     FMK00670
      IT=II+4                                                           FMK00680
      DO 200 L=II,IT,4                                                  FMK00690
      X1=CORDG(L,1)                                                     FMK00700
      Y1=CORDG(L,2)                                                     FMK00710
      Z1=CORDG(L,3)                                                     FMK00720
      W1(1,L)=D2(X,Y,Z,    Y1,Z1)                                       FMK00730
      W1(2,L)=D4(X,  Z,    Y1,Z1)                                       FMK00740
```

```
  200 W1(3,L)=D4(X,   Y,    Z1,Y1)                                          FMK00750
CC                                                                          FMK00760
CCC              (3) MIDSIDE NODES :    XI= + OR - 1                       FMK00770
CCC                                   ZETA= + OR - 1                       FMK00780
CCCC                                   ETA= 0                              FMK00790
CCCCC                                                                       FMK00800
CCCCC                           NODE # : 4,8,16,20                          FMK00810
C                                                                           FMK00820
      DO 300 K=1,2                                                          FMK00830
      II=(K-1)*12+4                                                         FMK00840
      IT=II+4                                                               FMK00850
      DO 300 L=II,IT,4                                                      FMK00860
      X1=CORDG(L,1)                                                         FMK00870
      Y1=CORDG(L,2)                                                         FMK00880
      Z1=CORDG(L,3)                                                         FMK00890
      W1(1,L)=D4(Y,   Z,    X1,Z1)                                          FMK00900
      W1(2,L)=D2(Y,Z,X,     Z1,X1)                                          FMK00910
  300 W1(3,L)=D4(Y,   X,    Z1,X1)                                          FMK00920
C                                                                           FMK00930
CCC              (4) MIDSIDE NODES :    XI= + OR - 1                       FMK00940
CCCC                                  ZETA= 0                               FMK00950
CCCCC                                  ETA= + OR - 1                        FMK00960
CCCCCC                                                                      FMK00970
CCCCCC                          NODE # : 9,10,11,12                         FMK00980
C                                                                           FMK00990
      DO 400 L=9,12                                                         FMK01000
      X1=CORDG(L,1)                                                         FMK01010
      Y1=CORDG(L,2)                                                         FMK01020
      Z1=CORDG(L,3)                                                         FMK01030
      W1(1,L)=D4(Z,   Y,    X1,Y1)                                          FMK01040
      W1(2,L)=D4(Z,   X,    Y1,X1)                                          FMK01050
  400 W1(3,L)=D2(Z,X,Y,     X1,Y1)                                          FMK01060
C                                                                           FMK01070
CC*************************************************************************FMK01080
C                                                                           FMK01090
      DO 500 I=1,3                                                          FMK01100
      DO 500 J=1,3                                                          FMK01110
      AJ(I,J)=0.0D0                                                         FMK01120
C                                                                           FMK01130
CC            FORMATION OF JACOBIAN MATRIX AND COMPUTATION OF DETERMINANT   FMK01140
C                                                                           FMK01150
      DO 500 K=1,NPEL                                                       FMK01160
  500 AJ(I,J)=AJ(I,J)+W1(I,K)*COREL(K,J)                                    FMK01170
      DTJ=AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2,1    FMK01180
     1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(3,2)*AJ(2,3)*AJ(1,1)-AJ(3,3)*    FMK01190
     2AJ(2,1)*AJ(1,2)                                                       FMK01200
C                                                                           FMK01210
CC            COMPUTATION OF DERIVATIVES OF SHAPE FUNCTIONS WITH RESPECT    FMK01220
```

```
C              TO GLOBAL COORDINATES                                          FMK01230
CCC                                                                           FMK01240
CCC            MINORS OF COEFFICIENT MATRIX                                   FMK01250
C                                                                             FMK01260
      AJIN(1,1)= (AJ(2,2)*AJ(3,3)-AJ(3,2)*AJ(2,3))/DTJ                        FMK01270
      AJIN(2,1)=-(AJ(2,1)*AJ(3,3)-AJ(3,1)*AJ(2,3))/DTJ                        FMK01280
      AJIN(3,1)= (AJ(2,1)*AJ(3,2)-AJ(3,1)*AJ(2,2))/DTJ                        FMK01290
      AJIN(1,2)=-(AJ(1,2)*AJ(3,3)-AJ(3,2)*AJ(1,3))/DTJ                        FMK01300
      AJIN(2,2)= (AJ(1,1)*AJ(3,3)-AJ(3,1)*AJ(1,3))/DTJ                        FMK01310
      AJIN(3,2)=-(AJ(1,1)*AJ(3,2)-AJ(3,1)*AJ(1,2))/DTJ                        FMK01320
      AJIN(1,3)= (AJ(1,2)*AJ(2,3)-AJ(2,2)*AJ(1,3))/DTJ                        FMK01330
      AJIN(2,3)=-(AJ(1,1)*AJ(2,3)-AJ(2,1)*AJ(1,3))/DTJ                        FMK01340
      AJIN(3,3)= (AJ(1,1)*AJ(2,2)-AJ(2,1)*AJ(1,2))/DTJ                        FMK01350
C                                                                             FMK01360
CCC           SOLUTION OF SYSTEM BY CRAMER'S RULE                            FMK01370
C                                                                             FMK01380
      DO 600 I=1,3                                                            FMK01390
      DO 600 J=1,NPEL                                                         FMK01400
      DNX(I,J)=0.0D0                                                          FMK01410
      DO 600 K=1,3                                                            FMK01420
  600 DNX(I,J)=DNX(I,J)+AJIN(I,K)*W1(K,J)                                     FMK01430
C                                                                             FMK01440
      DO 10 I=1,6                                                             FMK01450
      DO 10 J=1,NST                                                           FMK01460
      B1(J,I)=0.0D0                                                           FMK01470
   10 B(I,J)=0.0D0                                                            FMK01480
C                                                                             FMK01490
CC  FORMATION OF MATRIX  N*                                                   FMK01500
C                                                                             FMK01510
      DO 700 I=1,NPEL                                                         FMK01520
      I1=3*I-2                                                                FMK01530
      I2=I1+1                                                                 FMK01540
      I3=I2+1                                                                 FMK01550
      B(1,I1)=DNX(1,I)                                                        FMK01560
      B(2,I2)=DNX(2,I)                                                        FMK01570
      B(3,I3)=DNX(3,I)                                                        FMK01580
      B(4,I1)=DNX(2,I)                                                        FMK01590
      B(4,I2)=DNX(1,I)                                                        FMK01600
      B(5,I2)=DNX(3,I)                                                        FMK01610
      B(5,I3)=DNX(2,I)                                                        FMK01620
      B(6,I3)=DNX(1,I)                                                        FMK01630
  700 B(6,I1)=DNX(3,I)                                                        FMK01640
C                                                                             FMK01650
CCC                                                                           FMK01660
CCCC          THE NEXT SEGMENT IS NOT USED WHEN 'FORMK' IS CALLED BY          FMK01670
CCC           'STRESS'                                                        FMK01680
C                                                                             FMK01690
      IF(INDIC.EQ.2) RETURN                                                   FMK01700
```

```
C                                                                       FMK01710
CC              MULTIPLICATION OF MATRICES AND DETERMINANT OF JACOBIAN IN  FMK01720
CC              THE INTEGRANT OF STIFFNESS MATRIX FORMULA                 FMK01730
C                                                                       FMK01740
      DO 800 I=1,NST                                                    FMK01750
      DO 750 J=1,3                                                      FMK01760
      DO 750 K=1,3                                                      FMK01770
  750 B1(I,J)=B1(I,J)+B(K,I)*ELAST(K,J)                                 FMK01780
      DO 800 L=4,6                                                      FMK01790
  800 B1(I,L)=B1(I,L)+B(L,I)*ELAST(L,L)                                 FMK01800
      DO 900 I=1,NST                                                    FMK01810
      DO 900 J=I,NST                                                    FMK01820
      AK(I,J)=0.0D0                                                     FMK01830
      DO 900 K=1,6                                                      FMK01840
  900 AK(I,J)=AK(I,J)+B1(I,K)*B(K,J)                                    FMK01850
      DO 1000 I=1,NST                                                   FMK01860
      DO 1000 J=I,NST                                                   FMK01870
      AK(I,J)=AK(I,J)*DTJ                                               FMK01880
 1000 AK(J,I)=AK(I,J)                                                   FMK01890
C                                                                       FMK01900
      RETURN                                                            FMK01910
      END                                                               FMK01920
C
C
C*****!*****!*****!
```

```
C              BLOCKS ARE COUNTED ROW WISE                                      MRG00210
CC                                                                              MRG00220
C                                                                               MRG00230
      NTRK(I,J)=(I-1)*(MM)+J                                                    MRG00240
      ZRO=0.0D0                                                                 MRG00250
C                                                                               MRG00260
CCC           FILL ALL BLOCKS OF TOTAL STIFFNESS MATRIX IN ROW SEQUENCE         MRG00270
CCC           BY PICKING THE PROPER ELEMENTS FROM THE INDIVIDUAL                MRG00280
CCC           ELEMENT STIFFNESS MATRICES                                        MRG00290
C                                                                               MRG00300
      DO 400 I=1,NN                                                             MRG00310
      DO 400 J=1,MM                                                             MRG00320
      DO 100 I1=1,NS                                                            MRG00330
      DO 100 J1=1,NS                                                            MRG00340
  100 A(I1,J1)=ZRO                                                              MRG00350
C                                                                               MRG00360
CCC           EXAMINE ALL ELEMENT STIFFNESS MATRICES FOR TERMS BELONGING        MRG00370
CCC           TO THE BLOCK (I,J) AND TRANSFER THEM ACCORDINGLY                  MRG00380
CC                                                                              MRG00390
      REWIND 9                                                                  MRG00400
      DO 300 K=1,NEL                                                            MRG00410
      READ(9) NCON                                                              MRG00420
C                                                                               MRG00430
CCC           DETERMINE THE LOWEST AND HIGHEST NODE NUMBER FOR EACH             MRG00440
CCC           ELEMENT                                                           MRG00450
C                                                                               MRG00460
      DO 200 L=1,NPEL                                                           MRG00470
  200 LM(L)= NDF*NCON(L)-NDF                                                    MRG00480
      LMIN=5000                                                                 MRG00490
      LMAX=0                                                                    MRG00500
      DO 210 I2=1,NPEL                                                          MRG00510
      LMIN=MINO(LMIN,LM(I2))                                                    MRG00520
  210 LMAX=MAXO(LMAX,LM(I2))                                                    MRG00530
      LMIN=LMIN+1                                                               MRG00540
      LMAX=LMAX+1                                                               MRG00550
      NSI=NS*I                                                                  MRG00560
      NSIM=NSI-NS                                                               MRG00570
C                                                                               MRG00580
CCC           IF NO NODE OF THE ELEMENT EXAMINED BELONGS TO BLOCK (I,J)         MRG00590
CCC           GO TO THE NEXT ELEMENT                                            MRG00600
C                                                                               MRG00610
      IF(LMIN.GT.NSI) GO TO 300                                                 MRG00620
      IF(LMAX.LE.NSIM) GO TO 300                                                MRG00630
C                                                                               MRG00640
CCC           IF ONE OR MORE NODES OF THE ELEMENT EXAMINED BELONG TO            MRG00650
CCC           BLOCK (I,J) READ THE ELEMENT STIFFNESS MATRIX FROM DISK           MRG00660
C                                                                               MRG00670
      CALL RDISK1(K)                                                            MRG00680
```

```
C                                                                       MRG00690
CC            LOCATE THE ELEMENTS OF THE STIFFNESS MATRIX EXAMINED THAT MRG00700
CC            REFER TO BLOCK (I,J) AND TRANSFER THEM TO THE BLOCK       MRG00710
C                                                                       MRG00720
      DO 290 II=1,NPEL                                                  MRG00730
      DO 280 JJ=1,NPEL                                                  MRG00740
      DO 270 KK=1,NDF                                                   MRG00750
      III=LM(II)+KK-NS*(I-1)                                            MRG00760
      IF((III.GT.NS).OR.(III.LT.1)) GO TO 270                           MRG00770
      KKK=NDF*II-NDF+KK                                                 MRG00780
      DO 260 LL=1,NDF                                                   MRG00790
      JJJ=LM(JJ)+LL-NS*(J+I-2)                                          MRG00800
      IF((JJJ.GT.NS).OR.(JJJ.LT.1)) GO TO 260                           MRG00810
      LLL=NDF*JJ-NDF+LL                                                 MRG00820
      A(III,JJJ)=A(III,JJJ)+S(KKK,LLL)                                  MRG00830
  260 CONTINUE                                                          MRG00840
  270 CONTINUE                                                          MRG00850
  280 CONTINUE                                                          MRG00860
  290 CONTINUE                                                          MRG00870
  300 CONTINUE                                                          MRG00880
C                                                                       MRG00890
CC            WRITE THE COMPLETED BLOCK ON DISK                         MRG00900
C                                                                       MRG00910
      NTK=NTRK(I,J)                                                     MRG00920
      CALL WRDISK(NTK,B)                                                MRG00930
  400 CONTINUE                                                          MRG00940
C                                                                       MRG00950
CC            IF THE LAST BLOCK IS NOT COMPLETELY FILED REPLACE THE     MRG00960
CC            ZERO DIAGONAL ELEMENTS BY UNITS                           MRG00970
C                                                                       MRG00980
      IF(NN*NS.EQ.NEQ) GO TO 600                                        MRG00990
      NTK=NTRK(NN,1)                                                    MRG01000
      CALL RDDISK(NTK,B)                                                MRG01010
      DO 500 I=1,NS                                                     MRG01020
      IF(DABS(A(I,I)).LT.1.0D-14)  A(I,I)=1.0D0                         MRG01030
  500 CONTINUE                                                          MRG01040
      CALL WRDISK(NTK,B)                                                MRG01050
C                                                                       MRG01060
  600 CONTINUE                                                          MRG01070
      RETURN                                                            MRG01080
      END                                                               MRG01090
C
C
C*****************************************************************************
C*****************************************************************************
C
```

```
      SUBROUTINE FLOAD                                                      FLD00020
C                                                                           FLD00030
      IMPLICIT REAL*8(A-H,O-Z)                                              FLD00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MN,NS,NCOUNT,NST,NSTF,NCLD,      FLD00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 FLD00060
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(5)                              FLD00070
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),         FLD00080
     1RB3(60)                                                               FLD00090
      DIMENSION   A(60,60),S(60,60),R1(60),R2(60)                           FLD00100
      EQUIVALENCE(A(1,1),AK1(1,1)),(S(1,1),AK2(1,1)),(R1(1),RB1(1)),        FLD00110
     1(R2(1),RB2(1))                                                        FLD00120
C                                                                           FLD00130
      ZRO=0.000                                                             FLD00140
      REWIND 14                                                             FLD00150
      READ(14) CLOAD                                                        FLD00160
      INDEX=1                                                               FLD00170
      INCH=CLOAD(NDFP)*NDF                                                  FLD00180
C                                                                           FLD00190
CC            EXAMINE EACH RIGHT HAND BLOCK AND POSITION IN IT ALL          FLD00200
CC            CORRESPONDING NODAL LOAD VECTORS                              FLD00210
C                                                                           FLD00220
      DO 1100 I=1,NN                                                        FLD00230
      DO 100 J=1,NS                                                         FLD00240
  100 R1(J)=ZRO                                                             FLD00250
      IH=I*NS                                                               FLD00260
   50 IF(INCH.GT.IH) GO TO 1000                                             FLD00270
      IR1=INCH-(I-1)*NS-NDF                                                 FLD00280
      DO 150 K=1,NDF                                                        FLD00290
      IR=IR1+K                                                              FLD00300
  150 R1(IR)=CLOAD(K)                                                       FLD00310
C                                                                           FLD00320
CC            IF THE LOAD POSITIONED IN THE BLOCK IS THE LAST ONE STOP       FLD00330
CC            THE PROCESS                                                   FLD00340
C                                                                           FLD00350
      IF(INDEX.EQ.NCLD) GO TO 1200                                          FLD00360
C                                                                           FLD00370
      READ(14) CLOAD                                                        FLD00380
      INDEX=INDEX+1                                                         FLD00390
      INCH=CLOAD(NDFP)*NDF                                                  FLD00400
      GO TO 50                                                              FLD00410
C                                                                           FLD00420
CC            STORE COMPLETED BLOCK ON DISK                                 FLD00430
C                                                                           FLD00440
 1000 CALL WDSKR1(I)                                                        FLD00450
 1100 CONTINUE                                                              FLD00460
      RETURN                                                                FLD00470
 1200 CALL WDSKR1(I)                                                        FLD00480
      I1=I+1                                                                FLD00490
```

```
      IF(I1.GT.NN) RETURN                                                FLD00500
      DO 1300 K=I1,NN                                                    FLD00510
      DO 1350 L=1,NS                                                     FLD00520
 1350 R1(L)=ZRO                                                          FLD00530
      CALL WDSKR1(K)                                                     FLD00540
 1300 CONTINUE                                                           FLD00550
C                                                                        FLD00560
      RETURN                                                             FLD00570
      END                                                                FLD00580
C
C
C*********************************************************************
C*********************************************************************
C


      SUBROUTINE BCOND                                                   BC000020
C                                                                        BC000030
      IMPLICIT REAL*8(A-H,O-Z)                                           BC000040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,   BC000050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL              BC000060
      COMMON /NB2/ NCON(20),NBC(4)                                       BC000070
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                           BC000080
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),      BC000090
     1RB3(60)                                                            BC000100
      DIMENSION AKW1(3600),AKW2(3600)                                    BC000110
      EQUIVALENCE (AKW1(1),AK1(1,1)),(AKW2(1),AK2(1,1))                  BC000120
C                                                                        BC000130
CC           IN THE COMMENT STATEMENTS THROUGHOUT THIS SUBROUTINE THE    BC000140
CC           FOLLOWING DEFINITIONS APPLY :                               BC000150
CC                    READ : BRING INTO C.P.U. THE SPECIFIED QUANTITY    BC000160
CC                           FROM DIRECT ACCESS STORAGE                  BC000170
CC             BLOCK (I,J) : COEFFICIENT MATRIX BLOCK OF ROW 'I' AND     BC000180
CC                           COLUMN 'J'.                                 BC000190
CC                           ROW AND COLUMN IDENTIFICATIONS REFER TO     BC000200
CC                           THE MATRIX AS IS STORED (NN*MM). EXCEPTION  BC000210
CC                           IS MADE WHEN REFERING TO SYMMETRIC BLOCKS   BC000220
CC               R.H. VECTOR : RIGHT HAND SIDE VECTOR CORRESPONDING TO   BC000230
CC                           ROW 'I'                                     BC000240
CC                 REDUCTION : MODIFICATION OF RIGHT HAND VECTOR AND/OR  BC000250
CC                           COEFFICIENT BLOCK AS REQUIRED               BC000260
CC                 NTRK(I,J) : SEQUENCE NUMBER OF COEFFICIENT BLOCK (I,J) BC000270
CC                           COUNTED ROW-WISE IN THE NN*MM STIFFNESS     BC000280
CC                           MATRIX                                      BC000290
CC                                                                       BC000300
C                                                                        BC000310
      NTRK(I,J)=(I-1)*(MM)+J                                             BC000320
C                                                                        BC000330
```

```
C                                                                       BC000340
C*************************************************************************BC000350
C*************************************************************************BC000360
C                                                                       BC000370
C                                                                       BC000380
C                                                                       BC000390
C           THE FIRST PART OF SUBROUTINE 'BCOND' IS USED ONLY IN THE    BC000400
C           SOLUTION OF PROBLEMS WHICH DO NOT INCLUDE B.C.'S WITH       BC000410
C           PREDEFINED DISPLACEMENTS                                    BC000420
C                                                                       BC000430
      ABIGN=1.0D20                                                      BC000440
      REWIND 15                                                         BC000450
      READ(15) NBC                                                      BC000460
      NBC1=NBC(1)+NDF                                                   BC000470
      INDEX=1                                                           BC000480
C                                                                       BC000490
C           ALL DIAGONAL BLOCKS ARE EXAMINED AND REDUCED IF REQUIRED    BC000500
C           IN SEQUENCE                                                 BC000510
C                                                                       BC000520
      DO 150 I=1,NN                                                     BC000530
      IH=I*NS                                                           BC000540
      IL=(I-1)*NS+1                                                     BC000550
C                                                                       BC000560
C           IF THE B.C. TO BE APPLIED DOES NOT CORRESPOND TO THE        BC000570
C           DIAGONAL BLOCK (I,I) EXAMINE THE NEXT DIAGONAL BLOCK        BC000580
C                                                                       BC000590
  100 IF(NBC1.GT.IH) GO TO 150                                          BC000600
      NTK=NTRK(I,1)                                                     BC000610
      CALL RDDISK(NTK,AKW1)                                             BC000620
  110 IBL=NBC(1)+1-NS*(I-1)                                             BC000630
      IBH=IBL+NDF-1                                                     BC000640
      IC=1                                                              BC000650
      DO 120 K=IBL,IBH                                                  BC000660
       IC=IC+1                                                          BC000670
C                                                                       BC000680
C           IF THE B.C. CODE CORRESPONDING TO DIAGONAL ELEMENT (K,K)    BC000690
C           IS NOT '1' NO MODIFICATION IS REQUIRED                      BC000700
C                                                                       BC000710
      IF(NBC(IC).NE.1) GO TO 120                                        BC000720
      AK1(K,K)=AK1(K,K)+ABIGN                                           BC000730
  120 CONTINUE                                                          BC000740
C                                                                       BC000750
C           IF THE B.C. APPLIED IS THE LAST OF THE PROBLEM NO FURTHER   BC000760
C           MODIFICATION OF COEFFICIENT MATRIX IS REQUIRED              BC000770
C                                                                       BC000780
      IF(INDEX.EQ.NPBC) GO TO 130                                       BC000790
      READ(15) NBC                                                      BC000800
      NBC1=NBC(1)+NDF                                                   BC000810
```

```
      INDEX=INDEX+1                                                          BC000820
                                                                             BC000830
C                                                                            BC000840
CC           IF THE NEW B.C. DOES NOT CORRESPOND TO DIAGONAL BLOCK (I,I)    BC000850
CC           EXAMINE THE NEXT DIAGONAL BLOCK                                BC000860
C                                                                            BC000870
      IF(NBC1.LE.IH) GO TO 110                                               BC000880
      GO TO 140                                                              BC000890
  130 CALL WRDISK(NTK,AKW1)                                                  BC000900
      RETURN                                                                 BC000910
  140 CALL WRDISK(NTK,AKW1)                                                  BC000920
  150 CONTINUE                                                               BC000930
      RETURN                                                                 BC000940
C                                                                            BC000950
C*****************************************************************         BC000960
C*****************************************************************         BC000970
C                                                                            BC000980
CC                                                                           BC000990
CC           ENTRY 'BCOND2' IS USED ONLY FOR THE SOLUTION OF PROBLEMS       BC001000
CC           WHOSE B.C.'S INCORPORATE PREDEFINED DISPLACEMENTS              BC001010
CC                                                                           BC001020
C                                                                            BC001030
      ENTRY BCOND2                                                           BC001040
C                                                                            BC001050
      DIMENSION BCON1(60,2)                                                  BC001060
      EQUIVALENCE (BCON1(1,1),AK3(1,1))                                      BC001070
C                                                                            BC001080
      ABIGN=1.0D20                                                           BC001090
      I1=1                                                                   BC001100
      I5=0                                                                   BC001110
      MMF=NN-MM+1                                                            BC001120
C                                                                            BC001130
CC           DETERMINATION OF COLUMN 'IFR' AT WHICH THE FIRST B.C.          BC001140
CC           APPLIES                                                        BC001150
C                                                                            BC001160
      IFR=(NBCL*NDF+NS-1)/NS                                                 BC001170
C                                                                            BC001180
CC           DETERMINATION OF ROW 'IR1' AT WHICH REDUCTION MUST START       BC001190
C                                                                            BC001200
      IR1=1                                                                  BC001210
      IF(IFR.LE.MM) GO TO 160                                                BC001220
      IR1=IFR-MM+1                                                           BC001230
C                                                                            BC001240
C*****************************************************************         BC001250
C                                                                            BC001260
CC           REDUCTION OF ALL ROWS AND CORRESPONDING RIGHT HAND VECTORS     BC001270
CC           IN SEQUENCE STARTING WITH ROW 'IR1'                            BC001280
C                                                                            BC001290
```

```
C*****************************************************************BC001300
C                                                                 BC001310
  160 DO 900 I=IR1,NN                                             BC001320
      IFR1=IFR-I+1                                                BC001330
      IF(IFR1.GT.MM) GO TO 900                                    BC001340
      REWIND 15                                                   BC001350
      REWIND 22                                                   BC001360
C                                                                 BC001370
CC           READ THE RIGHT HAND VECTOR CORRESPONDING TO ROW 'I'  BC001380
C                                                                 BC001390
      NTKR1=I                                                     BC001400
      CALL RDSKR1(NTKR1)                                          BC001410
C                                                                 BC001420
CC           READ THE FIRST NOT APPLIED B.C.                      BC001430
C                                                                 BC001440
      DO 200 L=1,I1                                               BC001450
  200 READ(15) NBC                                                BC001460
      NBC1=NBC(1)+NDF                                             BC001470
      NBC2=NBC1/NDF                                               BC001480
C                                                                 BC001490
CCC          SET THE POINTER ON FILE 22 AT THE FIRST NOT YET APPLIED BC001500
CCC          PREDEFINED DISPLACEMENT                              BC001510
C                                                                 BC001520
      IF(I5.EQ.0) GO TO 203                                       BC001530
      DO 202 L=1,I5                                               BC001540
  202 READ(22) BCON                                               BC001550
C                                                                 BC001560
CC           DETERMINATION OF FIRST BLOCK OF ROW 'I' TO BE REDUCED AND BC001570
CC           TRANSFER OF CONTROL TO THE APPROPRIATE SEGMENT       BC001580
CCC                                                               BC001590
CCC                                                               BC001600
CCC          SELECT THE PROPER SEGMENT DEPENDING ON WHETHER THE FIRST BC001610
CCC          BLOCK TO BE REDUCED IS DIAGONAL OR NOT               BC001620
C                                                                 BC001630
  203 IFR1=IFR                                                    BC001640
      J3=2                                                        BC001650
      IF(IFR.NE.I) GO TO 500                                      BC001660
C                                                                 BC001670
C*****************************************************************BC001680
C                                                                 BC001690
CC                                                                BC001700
CC           REDUCTION OF DIAGONAL BLOCK AND RIGHT HAND BLOCK OF ROW 'I'BC001710
CC                                                                BC001720
C*****************************************************************BC001730
C                                                                 BC001740
      I4=0                                                        BC001750
      IH=I*NS                                                     BC001760
C                                                                 BC001770
```

```
C               READ THE 'I' DIAGONAL BLOCK                                     BC001780
C                                                                               BC001790
      NTK1=NTRK(I,1)                                                            BC001800
      CALL RDDISK(NTK1,AKW1)                                                    BC001810
C                                                                               BC001820
CC              EXAMINE THE B.C. CODES FOR NODE NBC2   AND PERFORM THE          BC001830
CC              NECESSARY MODIFICATIONS TO THE DIAGONAL BLOCK AND THE           BC001840
CC              RIGHT HAND VECTOR                                               BC001850
C                                                                               BC001860
      I61=0                                                                     BC001870
  205 I6=0                                                                      BC001880
      DO 250 K=1,NDF                                                            BC001890
      K1=K+1                                                                    BC001900
      M=NBC(K1)+1                                                               BC001910
      II=NBC(1)-(I-1)*NS+K                                                      BC001920
      GO TO (250,230,220),M                                                     BC001930
  220 IF(I6.NE.0) GO TO 222                                                     BC001940
      READ(22) BCON                                                             BC001950
      I5=I5+1                                                                   BC001960
      I6=1                                                                      BC001970
      I61=1                                                                     BC001980
  222 DO 225 L=1,NS                                                             BC001990
  225 RB1(L)=RB1(L)-AK1(L,II)*BCON(K)                                           BC002000
      I4=I4+1                                                                   BC002010
      BCON1(I4,1)=II                                                            BC002020
      BCON1(I4,2)=BCON(K)                                                       BC002030
  230 AK1(II,II)=AK1(II,II)+ABIGN                                               BC002040
  250 CONTINUE                                                                  BC002050
      J3=MM+1                                                                   BC002060
      MML=I/MMF-(I-MMF)/MMF                                                     BC002070
      MM1=MM-(I-MMF)*MML                                                        BC002080
      J4=MM1                                                                    BC002090
C                                                                               BC002100
CC              IF THE B.C. APPLIED IS THE LAST ONE, NO FURTHER REDUCTION       BC002110
CC              OF ROW 'I' IS REQUIRED                                          BC002120
CC              PROCEED WITH REDUCTION OF BLOCKS SYMMETRIC TO THE ONES OF       BC002130
CC              ROW 'I'                                                         BC002140
C                                                                               BC002150
      IF(NBC2.EQ.NBCH) GO TO 410                                                BC002160
C                                                                               BC002170
CC              READ THE NEXT B.C.                                              BC002180
C                                                                               BC002190
      READ(15) NBC                                                              BC002200
      NBC1=NBC(1)+NDF                                                           BC002210
      NBC2=NBC1/NDF                                                             BC002220
      I1=I1+1                                                                   BC002230
C                                                                               BC002240
C               IF THE NEW B.C.  DOES NOT CORRESPOND TO THE DIAGONAL BLOCK      BC002250
```

```
C             PROCEED WITH REDUCTION OF NON DIAGONAL BLOCKS AND THEIR     BC002260
CC            SYMMETRICS                                                 BC002270
C                                                                        BC002280
      IF(NBC1.LE.IH) GO TO 205                                           BC002290
C                                                                        BC002300
CC*************************************************************         BC002310
C                                                                        BC002320
C             AFTER COMPLETION OF REDUCTION OF DIAGONAL BLOCK THE        BC002330
C             FOLLOWING STEPS ARE EXECUTED                               BC002340
C               (1) DETERMINATION OF BLOCK 'J1' CORRESPONDING TO THE     BC002350
C                   LAST B.C. READ FROM DATA                             BC002360
C               (2) IF BLOCK 'J1' IS THE NEXT TO THE DIAGONAL, CONTROL   BC002370
C                   IS TRANSFERED TO THE SEGMENT REDUCING THE RIGHT HAND BC002380
C                   VECTORS CORRESPONDING TO NON DIAGONAL BLOCKS AND     BC002390
C                   THEIR SYMMETRICS                                     BC002400
C               (3) IF STEP (2) IS NOT EXECUTED REDUCE THE RIGHT HAND    BC002410
C                   VECTORS CORRESPONDING TO BLOCKS SYMMETRIC TO THE     BC002420
C                   ONES BETWEEN THE DIAGONAL AND 'J1'                   BC002430
C                                                                        BC002440
CC*************************************************************         BC002450
C                                                                        BC002460
CCC               STEP (1)                                               BC002470
C                                                                        BC002480
  400 J1=(NBC1+NS-1)/NS                                                  BC002490
      J2=I+1                                                             BC002500
      J3=J1-I+1                                                          BC002510
      IFR=J1                                                             BC002520
      IF(I61.EQ.0) GO TO 490                                             BC002530
      IH=J1*NS                                                           BC002540
C                                                                        BC002550
CCC               STEP (2)                                               BC002560
C                                                                        BC002570
      IF(J1.EQ.J2) GO TO 490                                             BC002580
C                                                                        BC002590
CCC               STEP (3)                                               BC002600
C                                                                        BC002610
      J4=J3-1                                                            BC002620
      IF(J4.GT.MM) J4=MM                                                 BC002630
  410 IF(I.EQ.NN) GO TO 950                                              BC002640
      DO 450 J=2,J4                                                      BC002650
      NTK2=NTRK(I,J)                                                     BC002660
      CALL RDDISK(NTK2,AKW2)                                             BC002670
      NTKR2=J+I-1                                                        BC002680
      CALL RDSKR2(NTKR2)                                                 BC002690
      DO 430 L1=1,I4                                                     BC002700
      IR=BCON1(L1,1)                                                     BC002710
      DO 430 L2=1,NS                                                     BC002720
  430 RB2(L2)=RB2(L2)-AK2(IR,L2)*BCON1(L1,2)                             BC002730
```

```
  450 CALL WDSKR2(NTKR2)                                                    BC002740
C                                                                           BC002750
CC*************************************************************************BC002760
C                                                                           BC002770
  490 CALL WRDISK(NTK1,AKW1)                                                BC002780
  500 IF(J3.GT.MM) GO TO 850                                                BC002790
      MML=I/MMF-(I-MMF)/MMF                                                 BC002800
      MM1=MM-(I-MMF)*MML                                                    BC002810
      DO 800 J=J3,MM1                                                       BC002820
      IH=(J+(I-1))*NS                                                       BC002830
C                                                                           BC002840
CC            READ BLOCK (I,J)                                              BC002850
C                                                                           BC002860
      NTK2=NTRK(I,J)                                                        BC002870
      CALL RDDISK(NTK2,AKW2)                                                BC002880
C                                                                           BC002890
      IHL=(J+I-2)*NS+1                                                      BC002900
      IPD=(IH+1-NBC1)*(IHL-NBC1)                                            BC002910
C             IF THE B.C. NBC2 DOES NOT CORRESPOND TO BLOCK (I,J)           BC002920
CC            PROCEED WITH REDUCTION OF ITS SYMMETRIC BLOCK                 BC002930
C                                                                           BC002940
                                                                            BC002950
      IF(IPD.GT.0) GO TO 600                                                BC002960
C             MODIFY RIGHT HAND VECTOR 'I' FOR ALL B.C.'S CORRESPONDING     BC002970
CC            TO BLOCK (I,J)                                                BC002980
C                                                                           BC002990
  510 I6=0                                                                  BC003000
      DO 560 K=1,NDF                                                        BC003010
      K1=K+1                                                                BC003020
      IF(NBC(K1).NE.2) GO TO 560                                            BC003030
      IF(I6.NE.0) GO TO 550                                                 BC003040
      READ(22) BCON                                                         BC003050
      I6=1                                                                  BC003060
  550 II=NBC(1)+K-(I+J-2)*NS                                                BC003070
      DO 555 L=1,NS                                                         BC003080
  555 RB1(L)=RB1(L)-AK2(L,II)*BCON(K)                                       BC003090
  560 CONTINUE                                                              BC003100
C                                                                           BC003110
CC            IF THE B.C. APPLIED IS THE LAST ONE, NO FURTHER REDUCTION     BC003120
CC            IN ROW 'I' IS REQUIRED                                        BC003130
CC            PROCEED WITH REDUCTION OF ITS SYMMETRIC BLOCK                 BC003140
C                                                                           BC003150
      IF(NBC2.EQ.NBCH) GO TO 600                                            BC003160
      READ(15) NBC                                                          BC003170
      NBC1=NBC(1)+NDF                                                       BC003180
      NBC2=NBC1/NDF                                                         BC003190
C                                                                           BC003200
CC            IF THE NEW B.C. DOES NOT CORRESPOND TO BLOCK (I,J) PROCEED    BC003210
```

```
C             WITH REDUCTION OF ITS SYMMETRIC BLOCK                                BC003220
C                                                                                  BC003230
      IF(NBC1.GT.IH) GO TO 600                                                     BC003240
      GO TO 510                                                                    BC003250
C                                                                                  BC003260
CC            MODIFY THE RIGHT HAND VECTOR CORRESPONDING TO THE BLOCK              BC003270
CC            SYMMETRIC TO BLOCK (I,J)                                             BC003280
CCC                                                                                BC003290
CCC                                                                                BC003300
CC            IF NO B.C.'S REFER TO THE DIAGONAL BLOCK (I,I) THEN NO               BC003310
CC            MODIFICATION OF RIGHT HAND VECTORS CORRESPONDING TO THE               BC003320
CC            SYMMETRIC OF BLOCK (I,J) IS REQUIRED                                 BC003330
C                                                                                  BC003340
  600 IFR6=IFR1*I61                                                                BC003350
      IF(IFR6.NE.I) GO TO 800                                                      BC003360
C                                                                                  BC003370
      NTKR2=J+I-1                                                                  BC003380
      CALL RDSKR2(NTKR2)                                                           BC003390
      DO 650 L1=1,I4                                                               BC003400
      IR=BCON1(L1,1)                                                               BC003410
      DO 650 L2=1,NS                                                               BC003420
  650 RB2(L2)=RB2(L2)-AK2(IR,L2)*BCON1(L1,2)                                       BC003430
      CALL WDSKR2(NTKR2)                                                           BC003440
  800 CONTINUE                                                                     BC003450
C                                                                                  BC003460
  850 CALL WDSKR1(NTKR1)                                                           BC003470
  900 CONTINUE                                                                     BC003480
      RETURN                                                                       BC003490
  950 CALL WDSKR1(NTKR1)                                                           BC003500
      CALL WRDISK(NTK1,AKW1)                                                       BC003510
C                                                                                  BC003520
      RETURN                                                                       BC003530
      END                                                                          BC003540
C
CC
C*****************************************************************************
C*****************************************************************************
C


      SUBROUTINE SOLVE                                                             SLV00020
C                                                                                  SLV00030
      IMPLICIT REAL*8(A-H,O-Z)                                                     SLV00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,             SLV00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                        SLV00060
      COMMON /B3/ AT1(60,60),AT2(60,60),AT3(60,60),RT1(60),RT2(60),                SLV00070
     1RT3(60)                                                                      SLV00080
      DIMENSION AK1(3600),AK2(3600),AK3(3600),RB1(60),RB2(60),RB3(60)              SLV00090
```

```
      EQUIVALENCE (AK1(1),AT1(1,1)),(AK2(1),AT2(1,1)),(AK3(1),AT3(1,1)),SLV00100
     1(RB1(1),RT1(1)),(RB2(1),RT2(1)),(RB3(1),RT3(1))                   SLV00110
                                                                        SLV00120
C            IN THE COMMENTS THROUGHOUT THIS SUBROUTINE THE FOLLOWING   SLV00130
C            DEFINITIONS APPLY :                                        SLV00140
C                                                                       SLV00150
C              REDUCTION : MODIFICATION OF THE QUANTITY SPECIFIED AS    SLV00160
C                          DICTATED BY THE ALGORITHM OF THE EQUATION    SLV00170
C                          SOLVER                                       SLV00180
C                 A(N,M) : BLOCK OF ROW 'N' AND COLUMN 'M' OF           SLV00190
C                          COEFFICIENT (STIFFNESS) MATRIX               SLV00200
C                          ROW AND COLUMN NUMBERS REFER TO THE (NN*MM)  SLV00210
C                          FORM OF THE STIFFNESS MATRIX                 SLV00220
C               A*(N,M) : INVERSE OF A(N,M)                             SLV00230
C               A'(N,M) : TRANSPOSE OF A(N,M)                           SLV00240
C                  B(N) : RIGHT HAND SIDE VECTOR OF ROW 'N'             SLV00250
C                  READ : TRANSFER OF SPECIFIED QUANTITY FROM DISK      SLV00260
C                         INTO CENTRAL PROCESSING UNIT                  SLV00270
C                 STORE : TRANSFER OF SPECIFIED QUANTITY FROM CENTRAL   SLV00280
C                         PROCESSING UNIT TO DISK                       SLV00290
C              NTRK(I,J) : SEQUENCE NUMBER OF COEFFICIENT BLOCK OF ROW  SLV00300
C                          'I' AND COLUMN 'J'                           SLV00310
C                          BLOCKS ARE COUNTED ROW-WISE                  SLV00320
C                                                                       SLV00330
      NTRK(I,J)=(I-1)*(MM)+J                                            SLV00340
C                                                                       SLV00350
C            ALL STEPS DESCRIBED BELOW ARE REPEATED FOR ALL ROWS OF THE SLV00360
C            AUGMENTED MATRIX OF THE SYSTEM                             SLV00370
C                                                                       SLV00380
      KMM=MM                                                            SLV00390
      N=0                                                               SLV00400
C*****************************************************************     SLV00410
C                                                                       SLV00420
C                                                                       SLV00430
C            REDUCTION OF ROW 'N'                                       SLV00440
C                                                                       SLV00450
  100 N=N+1                                                             SLV00460
C                                                                       SLV00470
C              (1) REDUCTION OF B(N)                                    SLV00480
C                                                                       SLV00490
C                                                                       SLV00500
C                  (1A) READ A(N,1)                                     SLV00510
C                                                                       SLV00520
C                       INVERT A(N,1)                                   SLV00530
C                                                                       SLV00540
      NTK=NTRK(N,1)                                                     SLV00550
      CALL RDDISK(NTK,AK2)                                              SLV00560
                                                                        SLV00570
```

```
      CALL SYMINV(AK2,NS,RB1,RB2,IFLG)                                  SLV00580
                                                                        SLV00590
C             (1B) IF A(N,1) IS SINGULAR GIVE ERROR MESSAGE AND         SLV00600
C                  STOP EXECUTION FOR THIS PROBLEM                      SLV00610
C                                                                       SLV00620
C                  IF A(N,1) IS NEARLY SINGULAR GIVE WARNING            SLV00630
C                  MESSAGE AND CONTINUE EXECUTION                       SLV00640
C                                                                       SLV00650
      IF(IFLG.EQ.1) CALL ERROR4(N)                                      SLV00660
      IF(IFLG.EQ.2) CALL ERROR5(N)                                      SLV00670
                                                                        SLV00680
C             (1C) READ B(N)                                            SLV00690
C                                                                       SLV00700
C                  MULTIPLY B(N) BY A*(N,1)                             SLV00710
C                                                                       SLV00720
C                  STORE THE PRODUCT IN PLACE OF B(N)                   SLV00730
C                                                                       SLV00740
      NTR=N                                                             SLV00750
      CALL RDSKR1(NTR)                                                  SLV00760
      CALL MULT(AK2,RB1,RB2,NS,NS,1)                                    SLV00770
      CALL WDSKR2(NTR)                                                  SLV00780
                                                                        SLV00790
C             (1D) IF ROW 'N' IS THE LAST ROW SKIP THE FOLLOWING        SLV00800
C                  STEPS AND START BACK SUBSTITUTION                    SLV00810
C                                                                       SLV00820
      IF(N.EQ.NN) GO TO 300                                             SLV00830
                                                                        SLV00840
C          (2) REDUCTION OF NON DIAGONAL COEFFICIENT BLOCKS OF          SLV00850
C              ROW 'N'                                                  SLV00860
C                                                                       SLV00870
C              IF BLANK BLOCKS EXIST SKIP THEM                          SLV00880
C                                                                       SLV00890
      IF(N.GT.(NN-MM+1)) KMM=KMM-1                                      SLV00900
C                                                                       SLV00910
C             THE FOLLOWING STEPS ARE REPEATED FOR ALL BLOCKS A(N,K     SLV00920
C                                                                       SLV00930
C                                                                       SLV00940
C             (2A) READ A(N,K)                                          SLV00950
C                                                                       SLV00960
C                  MULTIPLY A(N,K) BY A*(N,1)                           SLV00970
C                                                                       SLV00980
C                  STORE THE PRODUCT IN THE PLACE OF A(N,K)             SLV00990
C                                                                       SLV01000
      DO 200 K=2,KMM                                                    SLV01010
      NTK=NTRK(N,K)                                                     SLV01020
      CALL RDDISK(NTK,AK1)                                              SLV01030
      CALL MULT(AK2,AK1,AK3,NS,NS,NS)                                   SLV01040
      CALL WRDISK(NTK,AK3)                                              SLV01050
```

```
C                                                                       SLV01060
CCC                  (2B) COMPUTE A'(N,K)                               SLV01070
CCCCC                                                                   SLV01080
                          STORE A'NN,K) IN THE BLANK BLOCK A(NN,K)      SLV01090
C                                                                       SLV01100
      DO 150 I=1,NS                                                     SLV01110
      II=(I-1)*NS                                                       SLV01120
      DO 150 J=1,NS                                                     SLV01130
      IL=II+J                                                           SLV01140
      IR=I+(J-1)*NS                                                     SLV01150
  150 AK3(IL)=AK1(IR)                                                   SLV01160
      NTK=NTRK(NN,K)                                                    SLV01170
      CALL WRDISK(NTK,AK3)                                              SLV01180
  200 CONTINUE                                                          SLV01190
C                                                                       SLV01200
CCCCC            (3) REDUCE THE REMAINING ROWS STARTING WITH ROW 'N+1'  SLV01210
                                                                        SLV01220
                     THE FOLLOWING STEPS ARE REPEATED FOR EVERY ROW 'I' SLV01230
C                                                                       SLV01240
      DO 260 L=2,KMM                                                    SLV01250
      I=N+L-1                                                           SLV01260
      IF(I.GT.NN) GO TO 260                                             SLV01270
      J=0                                                               SLV01280
C                                                                       SLV01290
CCC                  (2A) READ A'(N,L) STORED IN BLOCK (NN,L)           SLV01300
C                                                                       SLV01310
      NTK=NTRK(NN,L)                                                    SLV01320
      CALL RDDISK (NTK,AK2)                                             SLV01330
C                                                                       SLV01340
CCCCC                (2B) THE FOLLOWING STEPS ARE REPEATED FOR ALL BLOCKS SLV01350
                          OF ROW 'I'                                    SLV01360
CCCCCCCCCCCCCCCCCCCCCCCCC                                               SLV01370
                          READ A(N,K)                                   SLV01380
                                                                        SLV01390
                          MULTIPLY A(N,K) BY A'(N,L)                    SLV01400
                                                                        SLV01410
                          READ A(I,J) AND SUBSTRACT FROM IT THE PRODUCT SLV01420
                          OF THE PREVIOUS STEP                          SLV01430
                                                                        SLV01440
                          STORE THE RESULT IN THE PLACE OF A(I,J)       SLV01450
C                                                                       SLV01460
      DO 250 K=L,KMM                                                    SLV01470
      J=J+1                                                             SLV01480
C                                                                       SLV01490
      NTK=NTRK(N,K)                                                     SLV01500
      CALL RDDISK(NTK,AK1)                                              SLV01510
      CALL MULT(AK2,AK1,AK3,NS,NS,NS)                                   SLV01520
C                                                                       SLV01530
```

```
      NTK=NTRK(I,J)                                                     SLV01540
      CALL RDDISK(NTK,AK1)                                              SLV01550
      DO 210 I1=1,NCOUNT                                                SLV01560
  210 AK1(I1)=AK1(I1)-AK3(I1)                                           SLV01570
      CALL WRDISK(NTK,AK1)                                              SLV01580
  250 CONTINUE                                                          SLV01590
C                                                                       SLV01600
CCC                   (2C) MULTIPLY B(N) BY A'(N,L)                     SLV01610
CCCCC                                                                   SLV01620
CCCCC                      READ B(I)                                    SLV01630
CCCCC                                                                   SLV01640
CCCCC                      FORM B(I)-B(N)*A'(N,L)                       SLV01650
CCCCC                                                                   SLV01660
CCCCC                      STORE THE PRODUCT IN THE PLACE OF B(I)       SLV01670
C                                                                       SLV01680
      CALL MULT(AK2,RB2,RB3,NS,NS,1)                                    SLV01690
      NTR=I                                                             SLV01700
      CALL RDSKR1(NTR)                                                  SLV01710
      DO 255 I1=1,NS                                                    SLV01720
  255 RB1(I1)=RB1(I1)-RB3(I1)                                           SLV01730
      CALL WDSKR1(NTR)                                                  SLV01740
  260 CONTINUE                                                          SLV01750
      GO TO 100                                                         SLV01760
C                                                                       SLV01770
C*************************************************************************SLV01780
CCC                                                                     SLV01790
CCC            BACK SUBSTITUTION                                        SLV01800
CCCCC                                                                   SLV01810
CCCCC          THE FOLLOWING STEPS ARE REPEATED FOR ALL ROWS STARTING   SLV01820
CCCCC          WITH ROW 'NN-1'                                          SLV01830
C                                                                       SLV01840
  300 N=N-1                                                             SLV01850
C                                                                       SLV01860
CCC            IF THE LAST ROW REDUCED WAS ROW NO 1 RETURN              SLV01870
CCC                                                                     SLV01880
      IF(N.EQ.0) GO TO 500                                              SLV01890
C                                                                       SLV01900
CCC            READ B(N)                                                SLV01910
CCC                                                                     SLV01920
      NT1=N                                                             SLV01930
      CALL RDSKR3(NT1)                                                  SLV01940
C                                                                       SLV01950
CCC            THE FOLLOWING ARE REPEATED FOR ALL BLOCKS A(N,K)         SLV01960
CCCCC                                                                   SLV01970
CCCCC            (1) READ A(N,K)                                        SLV01980
CCCCC                                                                   SLV01990
CCCCC            (2) READ B(L)  WHERE B(L)= VECTOR OF SOLUTIONS FOR THE SLV02000
C                    UNKNOWNS CORRESPONDING TO BLOCK A(N,K) FOUND IN    SLV02010
```

```
C                    PREVIOUS STEPS OF BACK SUBSTITUTION                    SLV02020
C                                                                           SLV02030
C                     (3) FORM B(NE-A(N,K)*B(L)                             SLV02040
C                                                                           SLV02050
C                   AT THE END OF THE OPERATIONS FOR ALL BLOCKS OF ROW 'N'  SLV02060
C                   THE VECTOR B(N) CONTAINS THE SOLUTIONS FOR THE UNKNOWNS SLV02070
C                   CORRESPONDING TO THE DIAGONAL BLOCK A(N,1) AND IS STORED SLV02080
C                   ON DISK                                                 SLV02090
C                                                                           SLV02100
      DO 400 K=2,KMM                                                        SLV02110
      L=N+K-1                                                               SLV02120
      IF(L.GT.NN) GO TO 400                                                 SLV02130
      NTK=NTRK(N,K)                                                         SLV02140
      CALL RDDISK(NTK,AK1)                                                  SLV02150
      NTR=L                                                                 SLV02160
      CALL RDSKR1(NTR)                                                      SLV02170
      CALL MULT(AK1,RB1,RB2,NS,NS,1)                                        SLV02180
      DO 310 K1=1,NS                                                        SLV02190
  310 RB3(K1)=RB3(K1)-RB2(K1)                                               SLV02200
  400 CONTINUE                                                              SLV02210
      CALL WDSKR3(NT1)                                                      SLV02220
      KMM=KMM+1                                                             SLV02230
      IF(KMM.GT.MM) KMM=MM                                                  SLV02240
      GO TO 300                                                             SLV02250
  500 RETURN                                                                SLV02260
      END                                                                   SLV02270
C                                                                           DSP00010
C                                                                           DSP00020
      SUBROUTINE DISP                                                       DSP00030
C                                                                           DSP00040
      IMPLICIT REAL*8(A-H,O-Z)                                              DSP00050
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,      DSP00060
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 DSP00070
      COMMON /NB2/ NCON(20),NBC(4)                                          DSP00080
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),         DSP00090
     1RB3(60)                                                               DSP00100
      DIMENSION REACT(4),REACT1(3)                                          DSP00110
      EQUIVALENCE (REACT(1),RB2(1)),(REACT(2),REACT1(1))                    DSP00120
C                                                                           DSP00130
      ABIGN=1.0D20                                                          DSP00140
      ZRO=0.0D0                                                             DSP00150
      REWIND 15                                                             DSP00160
      REWIND 17                                                             DSP00170
      READ(15) NBC                                                          DSP00180
      NBC1=NBC(1)+NDF                                                       DSP00190
      INDEX=1                                                               DSP00200
      NPBL=NS/NDF
```

```
      JLL=NDPT-(NDPT/NPBL)*NPBL                                          DSP00210
      IF(JLL.EQ.0)  JLL=NPBL                                             DSP00220
      NL1=0                                                              DSP00230
      WRITE(6,1000)                                                      DSP00240
C                                                                        DSP00250
CC          TRANSFER OF RIGHT HAND BLOCKS IN SEQUENCE FROM DISK INTO     DSP00260
CCC         CORE                                                         DSP00270
C                                                                        DSP00280
      DO 280 I=1,NN                                                      DSP00290
C                                                                        DSP00300
      CALL RDSKR1(I)                                                     DSP00310
      CALL WDISKB(I)                                                     DSP00320
C                                                                        DSP00330
CC          IF THE CONSTRAINED POINT EXAMINED IS NOT INCLUDED IN THE     DSP00340
CCC         RIGHT HAND BLOCK  'I'  SKIP NEXT SEGMENT                     DSP00350
C                                                                        DSP00360
      IM1=I-1                                                            DSP00370
      ICTI=IM1*NPBL                                                      DSP00380
      IH=I*NS                                                            DSP00390
      IL=IM1*NS+1                                                        DSP00400
  210 ICHECK=(NBC1-IH)*(IL-NBC1)                                         DSP00410
      IF(ICHECK.LT.0) GO TO 255                                          DSP00420
C                                                                        DSP00430
CC          DETERMINE THE REACTIONS AT THE CONSTRAINED POINTS AND        DSP00440
CCC         MODIFY CORRESPONDINGLY THE RIGHT HAND VECTOR                 DSP00450
C                                                                        DSP00460
  213 DO 215 J=1,NDFP                                                    DSP00470
  215 REACT(J)=ZRO                                                       DSP00480
      I1=NBC(1)-NS*(I-1)+1                                               DSP00490
      I2=I1+1                                                            DSP00500
      I3=I2+1                                                            DSP00510
C                                                                        DSP00520
      IF(NBC(2).EQ.0) GO TO 220                                          DSP00530
      REACT(2)=RB1(I1)*ABIGN                                             DSP00540
      RB1(I1)=ZRO                                                        DSP00550
  220 IF(NBC(3).EQ.0) GO TO 240                                          DSP00560
      REACT(3)=RB1(I2)*ABIGN                                             DSP00570
      RB1(I2)=ZRO                                                        DSP00580
  240 IF(NBC(4).EQ.0) GO TO 250                                          DSP00590
      REACT(4)=RB1(I3)*ABIGN                                             DSP00600
      RB1(I3)=ZRO                                                        DSP00610
  250 ANBC=NBC(1)/NDF+1                                                  DSP00620
      REACT(1)=ANBC                                                      DSP00630
      WRITE(17) REACT                                                    DSP00640
      IF(INDEX.EQ.NPBC) GO TO 255                                        DSP00650
      READ(15) NBC                                                       DSP00660
      NBC1=NBC(1)+NDF                                                    DSP00670
      INDEX=INDEX+1                                                      DSP00680
```

```
      GO TO 210                                                         DSP00690
C                                                                       DSP00700
  255 NL2=NL1+NPBL                                                      DSP00710
      JL=NPBL                                                           DSP00720
      IF(I.EQ.NN) JL=JLL                                                DSP00730
      IF(NL2.GT.NLINE) GO TO 262                                        DSP00740
      DO 260 M=1,JL                                                     DSP00750
      ICT=ICTI+M                                                        DSP00760
      K1=(M-1)*NDF+1                                                    DSP00770
      K2=K1+2                                                           DSP00780
  260 WRITE(6,2000) ICT,(RB1(N1),N1=K1,K2)                              DSP00790
      NL1=NL2                                                           DSP00800
      GO TO 280                                                         DSP00810
  262 J2=NLINE-NL1                                                      DSP00820
      DO 264 N=1,J2                                                     DSP00830
      ICT=ICTI+N                                                        DSP00840
      K1=(N-1)*NDF+1                                                    DSP00850
      K2=K1+2                                                           DSP00860
  264 WRITE(6,2000) ICT,(RB1(N2),N2=K1,K2)                              DSP00870
      NL1=NPBL-J2                                                       DSP00880
      WRITE(6,1000)                                                     DSP00890
      J1=J2+1                                                           DSP00900
      DO 266 M1=J1,JL                                                   DSP00910
      ICT=ICTI+M1                                                       DSP00920
      K1=(M1-1)*NDF+1                                                   DSP00930
      K2=K1+2                                                           DSP00940
  266 WRITE(6,2000) ICT,(RB1(N3),N3=K1,K2)                              DSP00950
  280 CONTINUE                                                          DSP00960
C                                                                       DSP00970
  285 SUMX=ZRO                                                          DSP00980
      SUMY=ZRO                                                          DSP00990
      SUMZ=ZRO                                                          DSP01000
      I2=ZRO                                                            DSP01010
      REWIND 17                                                         DSP01020
C                                                                       DSP01030
C             PRINT THE REACTIONS                                       DSP01040
C                                                                       DSP01050
      I1=1                                                              DSP01060
  290 I2=I2+NLINE                                                       DSP01070
      IF(NPBC.LE.I2) I2=NPBC                                            DSP01080
      WRITE(6,1100)                                                     DSP01090
      DO 300 L3=I1,I2                                                   DSP01100
      READ(17) REACT                                                    DSP01110
      ICT=REACT(1)                                                      DSP01120
```

```
      WRITE(6,2000) ICT,REACT1                                                  DSP01130
C                                                                               DSP01140
CC           CHECK THE EQUILIBRIUM                                              DSP01150
C                                                                               DSP01160
      SUMX=SUMX+REACT(2)                                                        DSP01170
      SUMY=SUMY+REACT(3)                                                        DSP01180
      SUMZ=SUMZ+REACT(4)                                                        DSP01190
  300 CONTINUE                                                                  DSP01200
      I1=I2+1                                                                   DSP01210
      IF(I2.LT.NPBC) GO TO 290                                                  DSP01220
      WRITE(6,3000) SUMX,SUMY,SUMZ                                              DSP01230
      RETURN                                                                    DSP01240
C                                                                               DSP01250
 1000 FORMAT(1H1,//,5X,'  D-I-S-P-L-A-C-E-M-E-N-T-S   ',                        DSP01260
     1           //,5X,5H N.PT,12X,3H  X,23X,2H Y,23X,2H Z,//)                  DSP01270
 1100 FORMAT(1H1,//,5X,'  R-E-A-C-T-I-O-N-S   ',                                DSP01280
     1           //,5X,5H N.PT,12X,3H  X,23X,2H Y,23X,2H Z,//)                  DSP01290
 2000 FORMAT(5X,I5,3(1PD25.16))                                                 DSP01300
 3000 FORMAT(///,' EQUILIBRIUM CHECK ',/,10X,3(1PD25.16))                       DSP01310
      END                                                                       DSP01320
C
CC
CC*****************************************************************************
CC*****************************************************************************
C


      SUBROUTINE STRESS(I)                                                      STR00020
C                                                                               STR00030
      IMPLICIT REAL*8(A-H,O-Z)                                                  STR00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,          STR00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                     STR00060
      COMMON /NB2/ NCON(20),NBC(4)                                              STR00070
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),             STR00080
     1SSNN1(7)                                                                  STR00090
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),             STR00100
     1RB3(60)                                                                   STR00110
      DIMENSION SNELM(20,6),UEL(60),SSELM(20,6),SSNN(6),B(6,60),SSSS(6),STR00120
     1CORDC(20,3),LM(20)                                                        STR00130
      EQUIVALENCE (AK3(1,1),B(1)),(AK3(1,13),SNELM(1)),(AK3(1,15),              STR00140
     1SSELM(1)),(AK3(1,17),SSNN(1)),(AK3(7,1),SSSS(1)),                         STR00150
     2(UEL(1),RB1(1))                                                           STR00160
      DATA CORDC/1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.0D0,            STR00170
     11.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,         STR00180
     21.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,           STR00190
     31.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,          STR00200
     4-1.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,             STR00210
     50.0D0,0.0D0,0.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,STR00220
```

```
     6-1.0D0,-1.0D0/                                                    STR00230
C                                                                       STR00240
      READ(9) NCON                                                      STR00250
      IF(KEL.EQ.0) GO TO 200                                            STR00255
      WRITE(6,1000) I                                                   STR00260
C                                                                       STR00270
CC            FORM THE VECTOR OF ELEMENT NODAL DISPLACEMENTS            STR00280
C                                                                       STR00290
  200 DO 300 J=1,NPEL                                                   STR00300
      LM(J)=NCON(J)                                                     STR00310
      JJ=LM(J)                                                          STR00320
      JEL=3*(J-1)                                                       STR00330
      DO 250 K=1,NDF                                                    STR00340
      CALL RDISKB(JJ)                                                   STR00350
      J2=JEL+K                                                          STR00360
  250 UEL(J2)=UJNT(K)                                                   STR00370
  300 CONTINUE                                                          STR00380
C                                                                       STR00390
CC            COMPUTATION OF ELEMENT NODAL STRAINS                      STR00400
C                                                                       STR00410
      DO 400 J=1,NPEL                                                   STR00420
      JJ=NCON(J)                                                        STR00430
      X=CORDG(J,1)                                                      STR00440
      Y=CORDG(J,2)                                                      STR00450
      Z=CORDG(J,3)                                                      STR00460
      CALL FORMK(X,Y,Z,2)                                               STR00470
      DO 310 J1=1,6                                                     STR00480
      SSSS(J1)=0.0D0                                                    STR00490
      SSNN(J1)=0.0D0                                                    STR00500
      DO 310 J2=1,NST                                                   STR00510
  310 SSNN(J1)=SSNN(J1)+B(J1,J2)*UEL(J2)                                STR00520
C                                                                       STR00530
CC            COMPUTATION OF NODAL STRESSES FROM NODAL STRAINS          STR00540
C                                                                       STR00550
      DO 320 J1=1,3                                                     STR00560
      DO 320 J2=1,3                                                     STR00570
  320 SSSS(J1)=SSSS(J1)+SSNN(J2)*ELAST(J1,J2)                           STR00580
      DO 330 K1=4,6                                                     STR00590
  330 SSSS(K1)=SSSS(K1)+SSNN(K1)*ELAST(K1,K1)                           STR00600
C                                                                       STR00610
CC            FORMATION OF NODAL STRESS AND STRAIN VECTORS FOR THIS     STR00620
CC            ELEMENT                                                   STR00630
C                                                                       STR00640
      CALL RDISKS(JJ)                                                   STR00650
      CALL RDISKN(JJ)                                                   STR00660
      SSSS1(7)=SSSS1(7)+1.0D0                                           STR00670
      SSNN1(7)=SSNN1(7)+1.0D0                                           STR00680
      DO 350 K=1,6                                                      STR00690
```

```
      SNELM(J,K)=SSNN(K)                                                    STR00700
      SSELM(J,K)=SSSS(K)                                                    STR00710
      SSSS1(K)=SSSS1(K)+SSSS(K)                                             STR00720
      SSNN1(K)=SSNN1(K)+SSNN(K)                                             STR00730
  350 CONTINUE                                                              STR00740
      CALL WDISKS(JJ)                                                       STR00750
      CALL WDISKN(JJ)                                                       STR00760
  400 CONTINUE                                                              STR00770
C                                                                           STR00780
CCC           IF PRINTING OF ELEMENT NODAL STRESSES AND STRAINS IS NOT      STR00790
CCC           REQUIRED SKIP OUTPUT SEGMENT                                  STR00800
C                                                                           STR00810
      IF(KEL.EQ.0) GO TO 500                                                STR00820
C                                                                           STR00830
CCC           OUTPUT                                                        STR00840
C                                                                           STR00850
      DO 450 K=1,NPEL                                                       STR00860
      WRITE(6,2000)  LM(K),(SNELM(K,L),L=1,6)                               STR00870
      WRITE(6,2000)  LM(K),(SSELM(K,M),M=1,6)                               STR00880
  450 WRITE(6,3000)                                                         STR00890
  500 CONTINUE                                                              STR00900
      RETURN                                                                STR00910
C                                                                           STR00920
 1000 FORMAT(1H1,//' S-T-R-A-I-N-S / S-T-R-E-S-S-E-S FOR ELEMENT ',I4,      STR00930
     1//,2X,'JOINT',6X,                                                     STR00940
     2'SNX/SSX',8X,'SNY/SSY',8X,'SNZ/SSZ',8X,'SNXY/SSXY',6X,'SNYZ/SSYZ',    STR00950
     36X,'SNZX/SSZX',//)                                                    STR00960
 2000 FORMAT(2X,I4,4X,6(1PD15.6))                                           STR00970
 3000 FORMAT(1H )                                                           STR00980
C                                                                           STR00990
      END                                                                   STR01000
C
C
C*****************************************************************************
C*****************************************************************************
C


      SUBROUTINE DISK                                                       DSK00020
C                                                                           DSK00030
      IMPLICIT REAL*8(A-H,O-Z)                                              DSK00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MI,NS,NCOUNT,NST,NSTF,NCLD,      DSK00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 DSK00060
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),         DSK00070
     1SSNN1(7)                                                              DSK00080
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),         DSK00090
     1RB3(60)                                                               DSK00100
      DIMENSION STK1(900),STK2(900),STK3(900),STK4(900),A(1)                DSK00110
```

```
      EQUIVALENCE (STK1(1),AK1(1,1)),(STK2(1),AK1(1,16)),                DSK00120
     1(STK3(1),AK1(1,31)),(STK4(1),AK1(1,46))                            DSK00130
C                                                                         DSK00140
C                                                                         DSK00190
      NCOUNT=NS*NS                                                        DSK00200
      LREC7=NCOUNT/NREC7                                                  DSK00210
      NS1=NS/NDF                                                          DSK00220
      RETURN                                                              DSK00230
C                                                                         DSK00240
C*************************************************************************DSK00250
C                                                                         DSK00260
      ENTRY WRDISK(NTRACK,A)                                              DSK00270
C                                                                         DSK00280
      I7=(NTRACK-1)*NREC7+1                                               DSK00290
      N1=1                                                                DSK00300
      DO 10 L=1,NREC7                                                     DSK00310
      N2=L*LREC7                                                          DSK00320
      WRITE(7'I7)            (A(J),J=N1,N2)                               DSK00330
   10 N1=N2+1                                                             DSK00340
      RETURN                                                              DSK00350
C                                                                         DSK00360
C*************************************************************************DSK00370
C                                                                         DSK00380
      ENTRY RDDISK(NTRACK,A)                                              DSK00390
C                                                                         DSK00400
      I7=(NTRACK-1)*NREC7+1                                               DSK00410
      N1=1                                                                DSK00420
      DO 20 L=1,NREC7                                                     DSK00430
      N2=L*LREC7                                                          DSK00440
      READ(7'I7,ERR=500)            (A(J),J=N1,N2)                        DSK00450
   20 N1=N2+1                                                             DSK00460
      RETURN                                                              DSK00470
C                                                                         DSK00480
C*************************************************************************DSK00490
C                                                                         DSK00500
      ENTRY  WDISK1(NTRACK)                                               DSK00510
C                                                                         DSK00520
      I8=(NTRACK-1)*4+1                                                   DSK00530
      WRITE(8'I8)    STK1                                                 DSK00540
      WRITE(8'I8)    STK2                                                 DSK00550
      WRITE(8'I8)    STK3                                                 DSK00560
      WRITE(8'I8)    STK4                                                 DSK00570
      RETURN                                                              DSK00580
C                                                                         DSK00590
C*************************************************************************DSK00600
C                                                                         DSK00610
      ENTRY  RDISK1(NTRACK,A)                                             DSK00620
C                                                                         DSK00630
```

```
      I8=(NTRACK-1)*4+1                                                 DSK00640
      READ(8'I8,ERR=500)    STK1                                        DSK00650
      READ(8'I8,ERR=500)    STK2                                        DSK00660
      READ(8'I8,ERR=500)    STK3                                        DSK00670
      READ(8'I8,ERR=500)    STK4                                        DSK00680
      RETURN                                                            DSK00690
C                                                                       DSK00700
C*****************************************************************    DSK00710
C                                                                       DSK00720
      ENTRY WDISKC(NTRACK)                                              DSK00730
C                                                                       DSK00740
      WRITE(10'NTRACK)    COORD                                         DSK00750
      RETURN                                                            DSK00760
C                                                                       DSK00770
C*****************************************************************    DSK00780
C                                                                       DSK00790
      ENTRY RDISKC(NTRACK)                                              DSK00800
C                                                                       DSK00810
      READ(10'NTRACK,ERR=500)       COORD                               DSK00820
      RETURN                                                            DSK00830
C                                                                       DSK00840
C*****************************************************************    DSK00850
C                                                                       DSK00860
      ENTRY RDSKR1(NTRACK)                                              DSK00870
C                                                                       DSK00880
      READ(20'NTRACK,ERR=500)    RB1                                    DSK00890
      RETURN                                                            DSK00900
C                                                                       DSK00910
C*****************************************************************    DSK00920
C                                                                       DSK00930
      ENTRY RDSKR2(NTRACK)                                              DSK00940
C                                                                       DSK00950
      READ(20'NTRACK,ERR=500)    RB2                                    DSK00960
      RETURN                                                            DSK00970
C                                                                       DSK00980
C*****************************************************************    DSK00990
C                                                                       DSK01000
      ENTRY RDSKR3(NTRACK)                                              DSK01010
C                                                                       DSK01020
      READ(20'NTRACK,ERR=500)    RB3                                    DSK01030
      RETURN                                                            DSK01040
C                                                                       DSK01050
C*****************************************************************    DSK01060
C                                                                       DSK01070
      ENTRY WDSKR1(NTRACK)                                              DSK01080
C                                                                       DSK01090
      WRITE(20'NTRACK)  RB1                                             DSK01100
      RETURN                                                            DSK01110
```

```
C                                                                       DSK01120
C*****************************************************************DSK01130
C                                                                       DSK01140
      ENTRY WDSKR2(NTRACK)                                              DSK01150
C                                                                       DSK01160
      WRITE(20'NTRACK)  RB2                                             DSK01170
      RETURN                                                            DSK01180
C                                                                       DSK01190
C*****************************************************************DSK01200
C                                                                       DSK01210
      ENTRY WDSKR3(NTRACK)                                              DSK01220
C                                                                       DSK01230
      WRITE(20'NTRACK)  RB3                                             DSK01240
      RETURN                                                            DSK01250
C                                                                       DSK01260
C*****************************************************************DSK01270
C                                                                       DSK01280
      ENTRY WDISKB(NTRACK)                                              DSK01290
C                                                                       DSK01300
      I21=(NTRACK-1)*NS1+1                                              DSK01310
      J1=1                                                              DSK01320
      J2=NDF                                                            DSK01330
      DO 600 I=1,NS1                                                    DSK01340
      WRITE(21'I21)     (RB1(J),J=J1,J2)                                DSK01350
      J1=J2+1                                                           DSK01360
      J2=J2+NDF                                                         DSK01370
  600 CONTINUE                                                          DSK01380
      RETURN                                                            DSK01390
C                                                                       DSK01400
C*****************************************************************DSK01410
C                                                                       DSK01420
      ENTRY RDISKB(NTRACK)                                              DSK01430
C                                                                       DSK01440
      READ (21'NTRACK,ERR=500)       UJNT                               DSK01450
      RETURN                                                            DSK01460
C                                                                       DSK01470
C*****************************************************************DSK01480
C                                                                       DSK01490
      ENTRY RDISKS(NTRACK)                                              DSK01500
C                                                                       DSK01510
      READ(12'NTRACK,ERR=500) SSSS1                                     DSK01520
      RETURN                                                            DSK01530
C                                                                       DSK01540
C*****************************************************************DSK01550
C                                                                       DSK01560
      ENTRY WDISKS(NTRACK)                                              DSK01570
C                                                                       DSK01580
      WRITE(12'NTRACK) SSSS1                                            DSK01590
```

```
      RETURN                                                                DSK01600
C                                                                           DSK01610
C*************************************************************************DSK01620
C                                                                           DSK01630
      ENTRY RDISKN(NTRACK)                                                  DSK01640
C                                                                           DSK01650
      READ(13'NTRACK,ERR=500) SSNN1                                         DSK01660
      RETURN                                                                DSK01670
C                                                                           DSK01680
C*************************************************************************DSK01690
C                                                                           DSK01700
      ENTRY WDISKN(NTRACK)                                                  DSK01710
C                                                                           DSK01720
      WRITE(13'NTRACK) SSNN1                                                DSK01730
      RETURN                                                                DSK01740
C                                                                           DSK01750
C*************************************************************************DSK01760
C                                                                           DSK01770
  500 CALL ERROR3                                                           DSK01780
      STOP                                                                  DSK01790
      END                                                                   DSK01800
C
C*************************************************************************
C*************************************************************************
C


      SUBROUTINE ERROR                                                      ERR00020
C                                                                           ERR00030
C             SUBROUTINE ERROR PERFORMS CHECKS AND/OR GIVES ERROR OR        ERR00040
C             WARNING MESSAGES WHEREVER REQUIRED                            ERR00050
C             IF EXECUTION OF THE PROBLEM IS STOPPED DUE TO AN ERROR        ERR00060
C             CONTROL IS TRANSFERED TO THE START OF THE EXECUTION OF        ERR00070
C             THE NEXT PROBLEM                                              ERR00080
C                                                                           ERR00090
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,     ERR00100
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 ERR00110
      COMMON/NB3/NR7,LRB7,LRW7,NNRC,NR8,LRB8,LRW8,NR9,LRB9,NR10,LRB10,      ERR00120
     1LRW10,NR12,LRB12,LRW12,NR13,LRB13,LRW13,NR14,LRB14,NR15,LRB15,NR16    ERR00130
     2,LRB16,NR17,LRB17,NR19,LRB19,NR20,LRB20,LRW20,NR21,LRB21,LRW21,       ERR00140
     3NR22,LRB22                                                            ERR00150
C                                                                           ERR00160
C*************************************************************************ERR00170
C                                                                           ERR00180
      ENTRY ERROR1(NMAT)                                                    ERR00190
C                                                                           ERR00200
C             ENTRY 'ERROR1' CHECKS ALL FILE CHARACTERISTICS APPEARING      ERR00210
```

```
C             IN 'DEFINE FILE' STATEMENTS AND CONTROL CARDS EXCEPT FOR    ERR00220
CCC           THE NUMBER OF RECORDS OF FILE NO 7 (SEE 'ERROR2')           ERR00230
CCC           THE NUMBER OF DIFFERENT MATERIALS IS ALSO CHECKED           ERR00240
C                                                                         ERR00250
      ISTOP=0                                                             ERR00260
      ISTP9=0                                                             ERR00270
      ISTP16=0                                                            ERR00280
C                                                                         ERR00290
CC            FILE NO 7 (EXCEPT NR7)                                      ERR00300
C                                                                         ERR00310
      IFILE=7                                                             ERR00320
      IF(LRB7-NCOUNT*8/NREC7) 40,60,50                                    ERR00330
   40 WRITE(6,2000)                                                       ERR00340
      WRITE(6,8000) IFILE                                                 ERR00350
      ISTOP=1                                                             ERR00360
      GO TO 60                                                            ERR00370
   50 WRITE(6,3000)                                                       ERR00380
      WRITE(6,8500) IFILE                                                 ERR00390
C                                                                         ERR00400
   60 IF(LRW7-NCOUNT*2/NREC7) 70,90,80                                    ERR00410
   70 WRITE(6,2000)                                                       ERR00420
      WRITE(6,9000) IFILE                                                 ERR00430
      ISTOP=1                                                             ERR00440
      GO TO 90                                                            ERR00450
   80 WRITE(6,3000)                                                       ERR00460
      WRITE(6,9500) IFILE                                                 ERR00470
C                                                                         ERR00480
   90 IF((LRW7*4-LRB7).EQ.0) GO TO 95                                     ERR00490
      WRITE(6,2000)                                                       ERR00500
      WRITE(6,9700) IFILE                                                 ERR00510
      ISTOP=1                                                             ERR00520
C                                                                         ERR00530
CC            FILE NO 8                                                   ERR00540
C                                                                         ERR00550
   95 IFILE=8                                                             ERR00560
      IF(NR8-NEL*4) 100,120,110                                           ERR00570
  100 WRITE(6,2000)                                                       ERR00580
      WRITE(6,6000) IFILE                                                 ERR00590
      ISTOP=1                                                             ERR00600
      GO TO 120                                                           ERR00610
  110 WRITE(6,3000)                                                       ERR00620
      WRITE(6,5000) IFILE                                                 ERR00630
C                                                                         ERR00640
  120 IF(LRB8-7200) 130,150,140                                           ERR00650
  130 WRITE(6,2000)                                                       ERR00660
      WRITE(6,8000) IFILE                                                 ERR00670
      ISTOP=1                                                             ERR00680
      GO TO 150                                                           ERR00690
```

```
  140 WRITE(6,3000)                                                    ERR00700
      WRITE(6,8500) IFILE                                              ERR00710
C                                                                      ERR00720
  150 IF(LRW8-1800) 160,180,170                                        ERR00730
  160 WRITE(6,2000)                                                    ERR00740
      WRITE(6,9000) IFILE                                              ERR00750
      ISTOP=1                                                          ERR00760
      GO TO 180                                                        ERR00770
  170 WRITE(6,3000)                                                    ERR00780
      WRITE(6,9500) IFILE                                              ERR00790
C                                                                      ERR00800
  180 IF((LRW8*4-LRB8).EQ.0) GO TO 185                                 ERR00810
      WRITE(6,2000)                                                    ERR00820
      WRITE(6,9700) IFILE                                              ERR00830
      ISTOP=1                                                          ERR00840
C                                                                      ERR00850
CC              FILE NO 9                                              ERR00860
C                                                                      ERR00870
  185 IFILE=9                                                          ERR00880
      IF(NR9-NEL) 190,210,200                                          ERR00890
  190 WRITE(6,2000)                                                    ERR00900
      WRITE(6,6000) IFILE                                              ERR00910
      ISTP9=1                                                          ERR00920
      GO TO 210                                                        ERR00930
  200 WRITE(6,3000)                                                    ERR00940
      WRITE(6,5000) IFILE                                              ERR00950
C                                                                      ERR00960
  210 IF(LRB9-80) 220,240,230                                          ERR00970
  220 WRITE(6,2000)                                                    ERR00980
      WRITE(6,8000) IFILE                                              ERR00990
      ISTP9=1                                                          ERR01000
      GO TO 240                                                        ERR01010
  230 WRITE(6,3000)                                                    ERR01020
      WRITE(6,8500) IFILE                                              ERR01030
C                                                                      ERR01040
CC              FILE NO 10                                             ERR01050
C                                                                      ERR01060
  240 IFILE=10                                                         ERR01070
      IF(NR10-NDPT) 250,270,260                                        ERR01080
  250 WRITE(6,2000)                                                    ERR01090
      WRITE(6,6000) IFILE                                              ERR01100
      ISTOP=1                                                          ERR01110
      GO TO 270                                                        ERR01120
  260 WRITE(6,3000)                                                    ERR01130
      WRITE(6,5000) IFILE                                              ERR01140
C                                                                      ERR01150
  270 IF(LRB10-24) 280,300,290                                         ERR01160
  280 WRITE(6,2000)                                                    ERR01170
```

```
      WRITE(6,8000) IFILE                                                ERR01180
      ISTOP=1                                                            ERR01190
      GO TO 300                                                          ERR01200
  290 WRITE(6,3000)                                                      ERR01210
      WRITE(6,8500) IFILE                                                ERR01220
C                                                                        ERR01230
  300 IF(LRW10-6) 310,330,320                                            ERR01240
  310 WRITE(6,2000)                                                      ERR01250
      WRITE(6,9000) IFILE                                                ERR01260
      ISTOP=1                                                            ERR01270
      GO TO 330                                                          ERR01280
  320 WRITE(6,3000)                                                      ERR01290
      WRITE(6,9500) IFILE                                                ERR01300
C                                                                        ERR01310
  330 IF((LRW10*4-LRB10).EQ.0) GO TO 335                                 ERR01320
      WRITE(6,2000)                                                      ERR01330
      WRITE(6,9700) IFILE                                                ERR01340
      ISTOP=1                                                            ERR01350
C                                                                        ERR01360
CC            FILE NO 12                                                 ERR01370
C                                                                        ERR01380
  335 IFILE=12                                                           ERR01390
      IF(NR12-NDPT) 340,360,350                                          ERR01400
  340 WRITE(6,2000)                                                      ERR01410
      WRITE(6,6000) IFILE                                                ERR01420
      ISTOP=1                                                            ERR01430
      GO TO 360                                                          ERR01440
  350 WRITE(6,3000)                                                      ERR01450
      WRITE(6,5000) IFILE                                                ERR01460
C                                                                        ERR01470
  360 IF(LPB12-56) 370,390,380                                           ERR01480
  370 WRITE(6,2000)                                                      ERR01490
      WRITE(6,8000) IFILE                                                ERR01500
      ISTOP=1                                                            ERR01510
      GO TO 390                                                          ERR01520
  380 WRITE(6,3000)                                                      ERR01530
      WRITE(6,8500) IFILE                                                ERR01540
C                                                                        ERR01550
  390 IF(LRW12-14) 400,420,410                                           ERR01560
  400 WRITE(6,2000)                                                      ERR01570
      WRITE(6,9000) IFILE                                                ERR01580
      ISTOP=1                                                            ERR01590
      GO TO 420                                                          ERR01600
  410 WRITE(6,3000)                                                      ERR01610
      WRITE(6,9500) IFILE                                                ERR01620
C                                                                        ERR01630
  420 IF((LRW12*4-LRB12).EQ.0) GO TO 425                                 ERR01640
      WRITE(6,2000)                                                      ERR01650
```

```
      WRITE(6,9700) IFILE                                                ERR01660
      ISTOP=1                                                            ERR01670
C                                                                        ERR01680
CC                  FILE NO 13                                           ERR01690
C                                                                        ERR01700
  425 IFILE=13                                                           ERR01710
      IF(NR13-NDPT) 430,450,440                                          ERR01720
  430 WRITE(6,2000)                                                      ERR01730
      WRITE(6,6000) IFILE                                                ERR01740
      ISTOP=1                                                            ERR01750
      GO TO 450                                                          ERR01760
  440 WRITE(6,3000)                                                      ERR01770
      WRITE(6,5000) IFILE                                                ERR01780
C                                                                        ERR01790
  450 IF(LRB13-56) 460,480,470                                           ERR01800
  460 WRITE(6,2000)                                                      ERR01810
      WRITE(6,8000) IFILE                                                ERR01820
      ISTOP=1                                                            ERR01830
      GO TO 480                                                          ERR01840
  470 WRITE(6,3000)                                                      ERR01850
      WRITE(6,8500) IFILE                                                ERR01860
C                                                                        ERR01870
  480 IF(LRW13-14) 490,510,500                                           ERR01880
  490 WRITE(6,2000)                                                      ERR01890
      WRITE(6,9000) IFILE                                                ERR01900
      ISTOP=1                                                            ERR01910
      GO TO 510                                                          ERR01920
  500 WRITE(6,3000)                                                      ERR01930
      WRITE(6,9500) IFILE                                                ERR01940
C                                                                        ERR01950
  510 IF((LRW13*4-LRB13).EQ.0) GO TO 515                                 ERR01960
      WRITE(6,2000)                                                      ERR01970
      WRITE(6,9700) IFILE                                                ERR01980
      ISTOP=1                                                            ERR01990
C                                                                        ERR02000
CC                  FILE NO 14                                           ERR02010
C                                                                        ERR02020
  515 IFILE=14                                                           ERR02030
      IF(NR14-NCLD) 520,540,530                                          ERR02040
  520 WRITE(6,2000)                                                      ERR02050
      WRITE(6,6000) IFILE                                                ERR02060
      ISTOP=1                                                            ERR02070
      GO TO 540                                                          ERR02080
  530 WRITE(6,3000)                                                      ERR02090
      WRITE(6,5000) IFILE                                                ERR02100
C                                                                        ERR02110
  540 IF(LRB14-32) 550,570,560                                           ERR02120
  550 WRITE(6,2000)                                                      ERR02130
```

```
      WRITE(6,8000) IFILE                                              ERR02140
      ISTOP=1                                                          ERR02150
      GO TO 570                                                        ERR02160
  560 WRITE(6,3000)                                                    ERR02170
      WRITE(6,8500) IFILE                                              ERR02180
C                                                                      ERR02190
CC            FILE NO 15                                               ERR02200
C                                                                      ERR02210
  570 IFILE=15                                                         ERR02220
      IF(NR15-NPBC) 580,600,590                                        ERR02230
  580 WRITE(6,2000)                                                    ERR02240
      WRITE(6,6000) IFILE                                              ERR02250
      ISTOP=1                                                          ERR02260
      GO TO 600                                                        ERR02270
  590 WRITE(6,3000)                                                    ERR02280
      WRITE(6,5000) IFILE                                              ERR02290
C                                                                      ERR02300
  600 IF(LRB15-16) 610,630,620                                         ERR02310
  610 WRITE(6,2000)                                                    ERR02320
      WRITE(6,8000) IFILE                                              ERR02330
      ISTOP=1                                                          ERR02340
      GO TO 630                                                        ERR02350
  620 WRITE(6,3000)                                                    ERR02360
      WRITE(6,8500) IFILE                                              ERR02370
C                                                                      ERR02380
CC            FILE NO 16                                               ERR02390
C                                                                      ERR02400
  630 IF(NMAT.EQ.1) GO TO 690                                          ERR02410
      IFILE=16                                                         ERR02420
      IF(NR16-NEL) 640,660,650                                         ERR02430
  640 WRITE(6,2000)                                                    ERR02440
      WRITE(6,6000) IFILE                                              ERR02450
      ISTOP=1                                                          ERR02460
      GO TO 660                                                        ERR02470
  650 WRITE(6,3000)                                                    ERR02480
      WRITE(6,5000) IFILE                                              ERR02490
C                                                                      ERR02500
  660 IF(LRB16-4) 670,690,680                                          ERR02510
  670 WRITE(6,2000)                                                    ERR02520
      WRITE(6,8000) IFILE                                              ERR02530
      ISTOP=1                                                          ERR02540
      GO TO 690                                                        ERR02550
  680 WRITE(6,3000)                                                    ERR02560
      WRITE(6,8500) IFILE                                              ERR02570
C                                                                      ERR02580
CC            FILE NO 17                                               ERR02590
C                                                                      ERR02600
  690 IFILE=17                                                         ERR02610
```

```
      IF(NR17-NPBC) 700,720,710                                         ERR02620
  700 WRITE(6,2000)                                                     ERR02630
      WRITE(6,6000) IFILE                                               ERR02640
      ISTOP=1                                                           ERR02650
      GO TO 720                                                         ERR02660
  710 WRITE(6,3000)                                                     ERR02670
      WRITE(6,5000) IFILE                                               ERR02680
C                                                                       ERR02690
  720 IF(LRB17-32) 730,750,740                                          ERR02700
  730 WRITE(6,2000)                                                     ERR02710
      WRITE(6,8000) IFILE                                               ERR02720
      ISTOP=1                                                           ERR02730
      GO TO 750                                                         ERR02740
  740 WRITE(6,3000)                                                     ERR02750
      WRITE(6,8500) IFILE                                               ERR02760
CCC                                                                     ERR02770
CCC           FILE NO 19                                                ERR02780
CCC                                                                     ERR02790
  750 IFILE=19                                                          ERR02800
      IF(NR19-NEL) 760,780,770                                          ERR02810
  760 WRITE(6,2000)                                                     ERR02820
      WRITE(6,6000) IFILE                                               ERR02830
      ISTOP=1                                                           ERR02840
      GO TO 780                                                         ERR02850
  770 WRITE(6,3000)                                                     ERR02860
      WRITE(6,5000) IFILE                                               ERR02870
C                                                                       ERR02880
  780 IF(LRB19-480) 790,810,800                                         ERR02890
  790 WRITE(6,2000)                                                     ERR02900
      WRITE(6,8000) IFILE                                               ERR02910
      ISTOP=1                                                           ERR02920
      GO TO 810                                                         ERR02930
  800 WRITE(6,3000)                                                     ERR02940
      WRITE(6,8500) IFILE                                               ERR02950
CCC                                                                     ERR02960
CCC           FILE NO 20                                                ERR02970
CCC                                                                     ERR02980
  810 IFILE=20                                                          ERR02990
      IF(NR20-NN) 820,840,830                                           ERR03000
  820 WRITE(6,2000)                                                     ERR03010
      WRITE(6,6000) IFILE                                               ERR03020
      ISTOP=1                                                           ERR03030
      GO TO 840                                                         ERR03040
  830 WRITE(6,3000)                                                     ERR03050
      WRITE(6,5000) IFILE                                               ERR03060
C                                                                       ERR03070
  840 IF(LPB20-NS*8) 850,870,860                                        ERR03080
  850 WRITE(6,2000)                                                     ERR03090
```

```
      WRITE(6,8000) IFILE                                                ERR03100
      ISTOP=1                                                            ERR03110
      GO TO 870                                                          ERR03120
  860 WRITE(6,2000)                                                      ERR03130
      WRITE(6,8500) IFILE                                                ERR03140
C                                                                        ERR03150
  870 IF(LRW20-NS*2) 880,900,890                                         ERR03160
  880 WRITE(6,2000)                                                      ERR03170
      WRITE(6,9000) IFILE                                                ERR03180
      ISTOP=1                                                            ERR03190
      GO TO 900                                                          ERR03200
  890 WRITE(6,3000)                                                      ERR03210
      WRITE(6,9500) IFILE                                                ERR03220
  900 IF((LRW20*4-LRB20).EQ.0) GO TO 905                                 ERR03230
      WRITE(6,2000)                                                      ERR03240
      WRITE(6,9700) IFILE                                                ERR03250
      ISTOP=1                                                            ERR03260
C                                                                        ERR03270
CC            FILE NO 21                                                 ERR03280
C                                                                        ERR03290
  905 IFILE=21                                                           ERR03300
      IF(NR21-NN*NS/NDF) 910,930,920                                     ERR03310
  910 WRITE(6,2000)                                                      ERR03320
      WRITE(6,6000) IFILE                                                ERR03330
      ISTOP=1                                                            ERR03340
      GO TO 930                                                          ERR03350
  920 WRITE(6,3000)                                                      ERR03360
      WRITE(6,5000) IFILE                                                ERR03370
C                                                                        ERR03380
  930 IF(LRB21-24) 940,960,950                                           ERR03390
  940 WRITE(6,2000)                                                      ERR03400
      WRITE(6,8000) IFILE                                                ERR03410
      ISTOP=1                                                            ERR03420
      GO TO 960                                                          ERR03430
  950 WRITE(6,3000)                                                      ERR03440
      WRITE(6,8500) IFILE                                                ERR03450
C                                                                        ERR03460
  960 IF(LRW21-6) 970,990,980                                            ERR03470
  970 WRITE(6,2000)                                                      ERR03480
      WRITE(6,9000) IFILE                                                ERR03490
      ISTOP=1                                                            ERR03500
      GO TO 990                                                          ERR03510
  980 WRITE(6,3000)                                                      ERR03520
      WRITE(6,9500) IFILE                                                ERR03530
  990 IF((LRW21*4-LRB21).EQ.0) GO TO 995                                 ERR03540
      WRITE(6,2000)                                                      ERR03550
      WRITE(6,9700) IFILE                                                ERR03560
      ISTOP=1                                                            ERR03570
```

```
C                                                                       ERR03580
CC            FILE NO 22                                                ERR03590
C                                                                       ERR03600
  995 IF(NPBC2.EQ.0) GO TO 1050                                         ERR03610
      IFILE=22                                                          ERR03620
      IF(NR22-NPBC2) 1000,1020,1010                                     ERR03630
 1000 WRITE(6,2000)                                                     ERR03640
      WRITE(6,6000) IFILE                                               ERR03650
      ISTOP=1                                                           ERR03660
      GO TO 1020                                                        ERR03670
 1010 WRITE(6,2000)                                                     ERR03680
      WRITE(6,5000) IFILE                                               ERR03690
C                                                                       ERR03700
 1020 IF(LRB22-24) 1030,1050,1040                                       ERR03710
 1030 WRITE(6,2000)                                                     ERR03720
      WRITE(6,8000) IFILE                                               ERR03730
      ISTOP=1                                                           ERR03740
      GO TO 1050                                                        ERR03750
 1040 WRITE(6,3000)                                                     ERR03760
      WRITE(6,8500) IFILE                                               ERR03770
C                                                                       ERR03780
CC            NUMBER OF MATERIALS                                       ERR03790
C                                                                       ERR03800
 1050 IF(NMAT.LE.10) GO TO 1055                                         ERR03810
      WRITE(6,2000)                                                     ERR03820
      WRITE(6,7000)                                                     ERR03830
      ISTOP=1                                                           ERR03840
C                                                                       ERR03850
 1055 IF((ISTP9.EQ.0).AND.(ISTP16.EQ.0)) RETURN                         ERR03860
      WRITE(6,3000)                                                     ERR03870
      WRITE(6,9600)                                                     ERR03880
      WRITE(6,4000)                                                     ERR03890
      CALL SKIP                                                         ERR03900
      CALL MAINE1                                                       ERR03910
C                                                                       ERR03920
C*****************************************************************    ERR03930
C                                                                       ERR03940
      ENTRY ERROR2                                                      ERR03950
C                                                                       ERR03960
CC            ENTRY 'ERROR2' CHECKS THE NUMBER OF RECORDS OF FILE NO 7  ERR03970
C                                                                       ERR03980
      IFILE=7                                                           ERR03990
      IF(NR7-MM*NN*NREC7) 1060,1080,1070                                ERR04000
 1060 WRITE(6,2000)                                                     ERR04010
      WRITE(6,6000) IFILE                                               ERR04020
      ISTOP=1                                                           ERR04030
      GO TO 1080                                                        ERR04040
 1070 WRITE(6,3000)                                                     ERR04050
```

```
      WRITE(6,5000) IFILE                                                   ERR04060
 1080 IF(ISTOP.EQ.0) RETURN                                                 ERR04070
      WRITE(6,4000)                                                         ERR04080
      CALL SKIP                                                             ERR04090
      CALL MAINE1                                                           ERR04100
C                                                                           ERR04110
CC*********************************************************************** ERR04120
C                                                                           ERR04130
      ENTRY ERROR3                                                          ERR04140
C                                                                           ERR04150
CC          ENTRY 'ERROR3' GIVES AN ERROR MESSAGE AND STOPS EXECUTION       ERR04160
CC          IF A MACHINE ERROR OCCURS DURING A 'READ DISK' OPERATION        ERR04170
C                                                                           ERR04180
      WRITE(6,2000)                                                         ERR04190
      WRITE(6,9800)                                                         ERR04200
      WRITE(6,4000)                                                         ERR04210
      CALL MAINE1                                                           ERR04220
C                                                                           ERR04230
CC*********************************************************************** ERR04240
C                                                                           ERR04250
      ENTRY ERROR4(M)                                                       ERR04260
C                                                                           ERR04270
CC          ENTRY 'ERROR4' GIVES AN ERROR MESSAGE AND STOPS EXECUTION       ERR04280
CC          IF A BLOCK OF THE STIFFNESS MATRIX IS SINGULAR                  ERR04290
C                                                                           ERR04300
      WRITE(6,3000)                                                         ERR04310
      WRITE(6,9900) M                                                       ERR04320
      WRITE(6,4000)                                                         ERR04330
      CALL MAINE1                                                           ERR04340
C                                                                           ERR04350
CC*********************************************************************** ERR04360
C                                                                           ERR04370
      ENTRY ERROR5(N)                                                       ERR04380
C                                                                           ERR04390
CC          ENTRY 'ERROR5' GIVES A WARNING MESSAGE IF A BLOCK OF THE        ERR04400
CC          STIFFNESS MATRIX IS NEARLY SINGULAR                             ERR04410
CC          EXECUTION IS NOT STOPPED                                        ERR04420
C                                                                           ERR04430
      WRITE(6,3000)                                                         ERR04440
      WRITE(6,9950) N                                                       ERR04450
      WRITE(6,4500)                                                         ERR04460
      RETURN                                                                ERR04470
C                                                                           ERR04480
CC*********************************************************************** ERR04490
C                                                                           ERR04500
 2000 FORMAT(////' ',38X,'****** E R R O R ******'//)                       ERR04510
 3000 FORMAT(////' ',35X,'****** W A R N I N G ******'//)                   ERR04520
 4000 FORMAT(/////' ',26X,'****** EXECUTION STOPS FOR THIS PROBLEM *****'   ERR04530
```

```
     1*'/////)                                                                  ERR04540
 4500 FORMAT(///' ',22X,'****** EXECUTION CONTINUES ******'////)               ERR04550
 5000 FORMAT(' ',11X,'THE SPECIFIED NUMBER OF RECORDS FOR FILE NO',I3,         ERR04560
     1' IS GREATER THAN THAT REQUIRED')                                         ERR04570
 6000 FORMAT(' ',13X,'THE SPECIFIED NUMBER OF RECORDS FOR FILE NO',I3,         ERR04580
     1' IS LESS THAN THAT REQUIRED')                                            ERR04590
 7000 FORMAT(' ',11X,'THE NUMBER OF DIFFERENT MATERIALS IS GREATER THAN         ERR04600
     1THE MAXIMUM PERMISSIBLE OF TEN')                                          ERR04610
 8000 FORMAT(' ',14X,'THE SPECIFIED RECORD LENGTH IN BYTES FOR FILE NO',        ERR04620
     1I3,' IS LESS THAN THAT REQUIRED')                                         ERR04630
 8500 FORMAT(' ',13X,'THE SPECIFIED RECORD LENGTH IN BYTES FOR FILE NO',        ERR04640
     1I3,' IS GREATER THAN THAT REQUIRED')                                      ERR04650
 9000 FORMAT(' ',14X,'THE SPECIFIED RECORD LENGTH IN WORDS FOR FILE NO',        ERR04660
     1I3,' IS LESS THAN THAT REQUIRED')                                         ERR04670
 9500 FORMAT(' ',13X,'THE SPECIFIED RECORD LENGTH IN WORDS FOR FILE NO',        ERR04680
     1I3,' IS GREATER THAN THAT REQUIRED')                                      ERR04690
 9600 FORMAT(////' ',1X,'THE NUMBER OF RECORDS OF FILE NO 7 HAS NOT BEEN        ERR04700
     1 CHECKED DUE TO ERROR IN CHAPACTERISTICS OF FILE NO 9 OR 16')             ERR04710
 9700 FORMAT(' ',15X,'SPECIFIED RECORD LENGTHS IN BYTES AND WORDS DO NOT        ERR04720
     1 AGREE FOR FILE NO',I3)                                                   ERR04730
 9800 FORMAT(' ',24X,'A MACHINE ERROR WAS MADE DURING A READ DISK OPERAT        ERR04740
     1ION')                                                                     ERR04750
 9900 FORMAT(' ',38X,'BLOCK NO',I3,' IS SINGULAR')                              ERR04760
 9950 FORMAT(' ',35X,'BLOCK NO',I3,' IS NEARLY SINGULAR')                       ERR04770
      STOP                                                                      ERR04780
      END                                                                       ERR04790
C
CC
CC*****************************************************************************
CC*****************************************************************************
C


      SUBROUTINE SKIP                                                           SKP00020
C                                                                               SKP00030
      IMPLICIT REAL*8(A-H,O-Z)                                                  SKP00040
      DATA START/'START   '/                                                    SKP00050
C                                                                               SKP00060
CC           IN CASE OF DETECTION OF AN ERROR BEFORE ALL INPUT DATA             SKP00070
CC           HAVE BEEN READ, SUBROUTINE 'SKIP' IS CALLED IN ORDER TO            SKP00080
CC           BY-PASS ALL DATA CARDS OF PROBLEM FOR WHICH EXEXCUTION             SKP00090
CC           STOPPED                                                            SKP00100
CC           AFTER COMPLETION OF THE BY-PASSING PROCESS THE FIRST DATA           SKP00110
CC           CARD IN THE READER IS THE FIRST CARD OF THE NEXT PROBLEM           SKP00120
C                                                                               SKP00130
   10 READ(5,1000) CHKWD                                                        SKP00140
      IF(CHKWD.NE.START) GO TO 10                                               SKP00150
 1000 FORMAT(1A8)                                                               SKP00160
```

```
      RETURN                                                                SKP00170
      END                                                                   SKP00180
C
CC*****************************************************************************
CC*****************************************************************************
C
C
      SUBROUTINE SORT                                                       SRT00020
C                                                                           SRT00030
      IMPLICIT REAL*8(A-H,O-Z)                                              SRT00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,      SRT00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 SRT00060
      COMMON /NB2/ NCON(20),NBC(4)                                          SRT00070
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(20,3),UJNT(3),SSSS1(7),         SRT00080
     1SSNN1(7)                                                              SRT00090
C                                                                           SRT00100
CCC           IN THIS SUBROUTINE THE ARRAYS OF ELEMENT NODAL COORDINATES    SRT00110
CCC           ' COREL(NPEL,NDF) ' ARE FORMED AND STORED ON DISK             SRT00120
C                                                                           SRT00130
      REWIND 9                                                              SRT00140
      REWIND 19                                                             SRT00150
      DO 300 I=1,NEL                                                        SRT00160
      READ(9) NCON                                                          SRT00170
      DO 200 J=1,NPEL                                                       SRT00180
      NTRACK=NCON(J)                                                        SRT00190
      CALL RDISKC(NTRACK)                                                   SRT00200
      DO 100 K=1,NDF                                                        SRT00210
      COREL(J,K)=COORD(K)                                                   SRT00220
  100 CONTINUE                                                              SRT00230
  200 CONTINUE                                                              SRT00240
      WRITE(19) COREL                                                       SRT00250
  300 CONTINUE                                                              SRT00260
      RETURN                                                                SRT00270
      END                                                                   SRT00280
C
CC*****************************************************************************
CC*****************************************************************************
C
C
      SUBROUTINE MULT(A,B,C,NRA,NCA,NCB)                                    MLT00020
      IMPLICIT REAL*8(A-H,O-Z)                                              MLT00030
      DIMENSION A(1),B(1),C(1)                                              MLT00040
      ZRO=0.0D0                                                             MLT00050
      DO 100 I=1,NRA                                                        MLT00060
```

```
      DO 100 J=1,NCB                                                    MLT00070
      IC=I+(J-1)*NRA                                                    MLT00080
      C(IC)=ZRO                                                         MLT00090
      DO 100 K=1,NCA                                                    MLT00100
      IA=I+(K-1)*NRA                                                    MLT00110
      IB=K+(J-1)*NCA                                                    MLT00120
      C(IC)=C(IC)+A(IA)*B(IB)                                           MLT00130
  100 CONTINUE                                                          MLT00140
      RETURN                                                            MLT00150
      END                                                               MLT00160
C
CC*****************************************************************************
CC*****************************************************************************
CC
      SUBROUTINE SYMINV(A,N,B,C,IFLG)                                   SMV00020
      IMPLICIT REAL*8(A-H,O-Z)                                          SMV00030
      DIMENSION A(1),B(1),C(1)                                          SMV00040
C*****************************************************************************SMV00050
CCCC                                                                    SMV00060
CCCC   SYMMETRICAL MATRIX INVERSION                                     SMV00070
CCCC                                                                    SMV00080
C*****************************************************************************SMV00090
      IFLG=0                                                            SMV00100
      ZRO=0.0D0                                                         SMV00110
      ABIGN=1.0D15                                                      SMV00120
      TRACE=ZRO                                                         SMV00130
      DO 5 I=1,N                                                        SMV00140
      IAD=(I-1)*N+I                                                     SMV00150
      IF(A(IAD).GE.ABIGN ) GO TO 5                                      SMV00160
      TRACE=TRACE+DABS(A(IAD))                                          SMV00170
    5 CONTINUE                                                          SMV00180
      APZRO=TRACE*1.0D-12                                               SMV00190
      IF(APZRO.GT.1.0D-20) APZRO=1.0D-20                                SMV00200
      LP=N                                                              SMV00210
      DO 10  I=2,LP                                                     SMV00220
      DO 10  J=I,LP                                                     SMV00230
      ICOL=J+LP*(I-2)                                                   SMV00240
      IROW=I+(J-1)*LP-1                                                 SMV00250
      IF(A(ICOL).EQ.A(IROW)) GO TO 10                                   SMV00260
      A(ICOL)=0.5D0*(A(ICOL)+A(IROW))                                   SMV00270
      A(IROW)=A(ICOL)                                                   SMV00280
   10 CONTINUE                                                          SMV00290
      DO 25 I=1,N                                                       SMV00300
      B(I)=ZRO                                                          SMV00310
      II=(I-1)*N                                                        SMV00320
```

```
      DO 20 J=1,N                                                       SMV00330
      JJ=II+J                                                           SMV00340
      IF(A( JJ).GE.ABIGN ) GO TO 20                                     SMV00350
      B(I)=B(I)+DABS(A(JJ))                                             SMV00360
   20 CONTINUE                                                          SMV00370
   25 CONTINUE                                                          SMV00380
      ANR=ZRO                                                           SMV00390
      DO 30 I=1,N                                                       SMV00400
   30 ANR=DMAX1(ANR,B(I))                                               SMV00410
      DO 145 I=1,N                                                      SMV00420
      NR=(I-1)*N                                                        SMV00430
C                                                                       SMV00440
      DO 100 J=1,N                                                      SMV00450
      K=NR+J                                                            SMV00460
  100 B(J)=A(K)                                                         SMV00470
      D=B(I)                                                            SMV00480
      IF(D.EQ.ZRO) GO TO 180                                            SMV00490
      IF(DABS(D).LT.APZRO) IFLG=2                                       SMV00500
      DO 110 J=1,N                                                      SMV00510
  110 C(J)=-B(J)/D                                                      SMV00520
C                                                                       SMV00530
      L=1                                                               SMV00540
      DO 130 J=1,N                                                      SMV00550
      M=L                                                               SMV00560
      DO 120 K=J,N                                                      SMV00570
      DB=DABS(B(J))                                                     SMV00580
      IF(DB.LE.1.0D-40) GO TO 115                                       SMV00590
      DC=DABS(C(K))                                                     SMV00600
      IF(DC.LE.1.0D-40) GO TO 115                                       SMV00610
      IF((DB.LE.APZRO).AND.(DC.LE.APZRO)) GO TO 115                     SMV00620
      A(L)=A(L)+B(J)*C(K)                                               SMV00630
  115 CONTINUE                                                          SMV00640
      A(M)=A(L)                                                         SMV00650
      M=M+N                                                             SMV00660
  120 L=L+1                                                             SMV00670
  130 L=L+J                                                             SMV00680
C                                                                       SMV00690
      C(I)=-1.0D0/D                                                     SMV00700
      M=I                                                               SMV00710
      DO 140 J=1,N                                                      SMV00720
      K=NR+J                                                            SMV00730
      A(K)=C(J)                                                         SMV00740
      A(M)=C(J)                                                         SMV00750
  140 M=M+N                                                             SMV00760
  145 CONTINUE                                                          SMV00770
C                                                                       SMV00780
      NS=N*N                                                            SMV00790
      DO 150 J=1,NS                                                     SMV00800
```

```
  150 A(J)=-A(J)                                                        SMV00810
C                                                                       SMV00820
      DO 165 I=1,N                                                      SMV00830
      B(I)=ZRO                                                          SMV00840
      II=(I-1)*N                                                        SMV00850
      DO 160 J=1,N                                                      SMV00860
      JJ=II+J                                                           SMV00870
      IF(A(JJ).EQ.1.0D0) GO TO 160                                      SMV00880
      B(I)=B(I)+DABS(A(JJ))                                             SMV00890
  160 CONTINUE                                                          SMV00900
  165 CONTINUE                                                          SMV00910
      AINR=ZRO                                                          SMV00920
      DO 170 I=1,N                                                      SMV00930
  170 AINR=DMAX1(AINR,B(I))                                             SMV00940
      IF(AINR.LT.1.0D-15) AINR=1.0D0/ANR                                SMV00950
      CNBR=ANR*AINR                                                     SMV00960
      WRITE(6,2000) CNBR,ANR,AINR                                       SMV00970
 2000 FORMAT(5X,'CONDITION NUMBER ',5X,1PD25.16,5X,2(1PD25.16))         SMV00980
      RETURN                                                            SMV00990
  180 IFLG=1                                                            SMV01000
      RETURN                                                            SMV01010
      END                                                               SMV01020
```

```
                                                                      MAI00010
                                                                      MAI00015
                                                                      MAI00020
                                                                      MAI00025
C             CONTROL PROGRAM                                         MAI00030
C
C                    *                    *
C                    **                   **
C                    ***                  ***
C             SAME AS FOR QUADRATIC ELEMENTS
C                    ***                  ***
C                    **                   **
C                    *                    *
C                                                                     MAI00200
C                                                                     MAI00210
      STOP
      END
C
C*****************************************************************************
C*****************************************************************************
C
                                                                      SMA00020
                                                                      SMA00030
      SUBROUTINE MAINE                                                SMA00040
C                                                                     SMA00050
      IMPLICIT REAL*8(A-H,O-Z)                                        SMA00060
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD, SMA00070
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDF2,NLINE,NREC7,KEL           SMA00080
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(32,3),UJNT(3),SSSS1(7),
     1SSNN1(7)
C                    *                    *
C                    **                   **
C                    ***                  ***
C             SAME AS FOR QUADRATIC ELEMENTS
C                    ***                  ***
C                    **                   **
C                    *                    *
C                                                                     SMA02640
      END
C
C*****************************************************************************
C*****************************************************************************
C
                                                                      INP00020
                                                                      INP00030
      SUBROUTINE INPUT                                                INP00040
C                                                                     INP00050
      IMPLICIT REAL*8(A-H,O-Z)                                        INP00060
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDF2,NLINE,NREC7,KEL
```

```
      COMMON /NB2/ NCON(32),NBC(4)                                          INP00070
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                              INP00080
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(32,3),UJNT(3),SSSS1(7),         INP00090
     1SSNN1(7)                                                              INP00100
      DIMENSION TITLE(10),FMT(10),CLOAD1(3),NCON1(16),NCON2(16)             INP00110
      EQUIVALENCE (CLOAD(1),CLOAD1(1)),(TITLE(1),COREL(1,1)),               INP00120
     1(FMT(1),COREL(1,3)),(NCON(1),NCON1(1)),(NCON(17),NCON2(1))            INP00130
C                     *                       *
C                     *                       *
C                     *                       *
C            SAME AS FOR QUADRATIC ELEMENTS
C                     *                       *
C                     *                       *
C                     *                       *
C                                                                           INP00500
C            CONNECTIVITY MATRIX                                            INP00510
C                                                                           INP00520
      READ(5,1000) TITLE                                                    INP00530
      WRITE(6,1100)                                                         INP00540
      WRITE(6,1000) TITLE                                                   INP00550
      READ(5,1000) TITLE                                                    INP00555
      WRITE(6,1000) TITLE                                                   INP00560
      WRITE(6,1100)                                                         INP00565
      READ(5,1000) FMT                                                      INP00570
      NBAND=0                                                               INP00580
C                                                                           INP00590
C            SELECT THE APPROPRIATE SEGMENT (1) OR (2) DEPENDING ON         INP00600
C            WHETHER ONE OR MORE MATERIALS  ARE PRESENT IN THE STRUCTURE     INP00610
C                                                                           INP00620
      IF(NMAT.EQ.1) GO TO 150                                               INP00630
C                                                                           INP00640
C              (1) THIS SEGMENT IS USED FOR THE CASE OF MORE THEN ONE       INP00650
C                  MATERIALS PRESENT IN THE STRUCTURE                       INP00660
C                                                                           INP00670
      DO 100 I=1,NEL                                                        IMP00680
      READ(5,FMT) NCON1                                                     IMP00685
      WRITE(6,FMT) NCON1                                                    INP00690
      READ(5,FMT) NCON2,NCON21                                              INP00695
      WRITE(6,FMT) NCON2,NCON21                                             INP00700
      WRITE(9) NCON                                                         INP00710
      WRITE(16) NCON21                                                      INP00720
C                                                                           INP00730
C            COMPUTATION OF HALF BANDWIDTH                                  INP00740
C                                                                           INP00750
      NPELM=NPEL-1                                                          INP00760
      DO 100 J=1,NPELM                                                      INP00770
      JP=J+1                                                                INP00780
      DO 100 K=JP,NPEL                                                      INP00790
  100 NBAND=MAX0(NBAND,IABS(NCON(J)-NCON(K)))
```

```
      GO TO 200                                                          INP00800
C                                                                        INP00810
CCC                                                                      INP00820
CCC             (2) THIS SEGMENT IS USED FOR THE CASE OF ONE MATERIAL    INP00830
C                                                                        INP00840
  150 DO 160 I=1,NEL                                                     INP00850
      READ(5,FMT) NCON1                                                  INP00860
      WRITE(6,FMT) NCON1                                                 INP00865
      READ(5,FMT) NCON2                                                  INP00870
      WRITE(6,FMT) NCON2                                                 INP00875
      WRITE(9) NCON                                                      INP00880
C                                                                        INP00890
CCC           COMPUTATION OF HALF BANDWIDTH                              INP00900
C                                                                        INP00910
      NPELM=NPEL-1                                                       INP00920
      DO 160 J=1,NPELM                                                   INP00930
      JP=J+1                                                             INP00940
      DO 160 K=JP,NPEL                                                   INP00950
  160 NBAND=MAXO(NBAND,IABS(NCON(J)-NCON(K)))                            INP00960
C                                                                        INP00970
  200 NBAND=(NBAND+1)*NDF                                                INP00980
      MM=(NBAND+2*(NS-1))/NS                                             INP00990
      IF(NN.LT.MM) MM=NN                                                 INP01000
       WRITE(6,4000) NEQ,NBAND,NN,MM,NCOUNT,NSTF                         INP01010
C                     *                    *
CC                    *                    *
CCC                   *                    *
CCCC          SAME AS FOR QUADRATIC ELEMENTS
CCCCC                 *                    *
CCCC                  *                    *
CCC                   *                    *
      END                                                                INP02390
C
CC
C*************************************************************************
C*************************************************************************
C


      SUBROUTINE ELPROP(I)                                               ELP00020
      IMPLICIT REAL*8(A-H,O-Z)                                           ELP00030
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                           ELP00040
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(32,3),UJNT(3),SSSS1(7),      ELP00050
     1SSNN1(7)                                                           ELP00060
C                     *                    *
CC                    *                    *
CCC                   *                    *
C             SAME AS FOR QUADRATIC ELEMENTS
```

```
C                     *                     *
C                     *                     *
C                     *                     *
      END                                                              ELP00280
C
C
C*****************************************************************************
C*****************************************************************************
C
C
      SUBROUTINE FSTF                                                  FST00020
C                                                                      FST00030
      IMPLICIT REAL*8(A-H,O-Z)                                         FST00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD, FST00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL            FST00060
      COMMON /B2/ ELAST(6,6),COORD(3),CORE(32,3),UJNT(3),SSSS1(7),     FST00070
     1SSNN1(7)                                                         FST00080
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),    FST00090
     1RB3(96)                                                          FST00100
      DIMENSION STK(9216),AK(9216),B(576)                              FST00110
C                     *                     *
C                     *                     *
C                     *                     *
C             SAME AS FOR QUADRATIC ELEMENTS
C                     *                     *
C                     *                     *
C                     *                     *
      END                                                              FST00460
C
C
C*****************************************************************************
C*****************************************************************************
C
C
      SUBROUTINE CUB
C
C
C     LISTINGS OF SUBROUTINES PERFORMING NUMERICAL INTEGRATION BY USE OF
C     TWO, THREE, FOUR, OR FIVE GAUSS POINTS ARE SHOWN IN APPENDIX
C
C
C
C*****************************************************************************
```

```
C*******************************************************************************
C
      SUBROUTINE FORMK(X,Y,Z,INDIC)                                        FMK00020
                                                                           FMK00030
C                                                                          FMK00040
      IMPLICIT REAL*8(A-H,O-Z)                                             FMK00050
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,     FMK00060
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                FMK00070
      COMMON /NB2/ NCON(32),NBC(4),LM(32)                                  FMK00080
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(32,3)                          FMK00090
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),        FMK00100
     1RB3(96)                                                              FMK00110
      DIMENSION AJ(3,3),AJIN(3,3),DNX(3,32),W1(3,32),B(6,96),B1(96,6),     FMK00120
     2AK(96,96),CORDG(32,3),STK(96,96)                                     FMK00130
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),          FMK00140
     1AK3(1,1)),(B1(1,1),AK3(1,7)),(W1(1,1),RB3(1))                        FMK00150
C                                                                          FMK00160
      DATA CORDG/1.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,      FMK00170
     1-1.0D0,-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,   FMK00180
     21.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0),1.0D0,-1.0D0,-1.0D0,1.0D0,    FMK00190
     31.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,-1.0D0,-1.0D0,   FMK00200
     4-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,1.0D0,1.0D0,     FMK00210
     51.0D0,1.0D0,1.0D0,1.0D0,0.3333333333333333,-0.3333333333333333,      FMK00220
     6-1.0D0,-1.0D0,-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333,  FMK00230
     7,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,    FMK00240
     81.0D0,1.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,-1.0D0,    FMK00250
     9-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,1.0D0,    FMK00260
     A1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,         FMK00270
     B0.3333333333333333,0.3333333333333333,0.3333333333333333,0.3333333  FMK00280
     C333333333,-0.3333333333333333,-0.3333333333333333 ,-0.333333333333  FMK00290
     D3333,-0.3333333333333333,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0  FMK00295
     E,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0/                          FMK00300
C                                                                          FMK00305
C*******************************************************************************
C                                                                          FMK00310
C                                                                          FMK00320
C           GENERAL FORMULAE FOR COMPUTATION OF DERIVATIVES OF SHAPE       FMK00330
C           FUNCTIONS WITH RESPECT TO 'XI','ETA','ZETA'                    FMK00340
C                                                                          FMK00350
      DFX(X,Y,Z,X1,Y1,Z1)=(1.0D0+Y*Y1)*(1.0D0+Z*Z1)*(9.0D0*X1*(X*X+Y*Y+    FMK00360
     1Z*Z)-19.0D0*X1+18.0D0*X+18.0D0*X*X*X1)/64.0D0                        FMK00370
      DX(X,Y,Z,X1,Y1,Z1)=9.0D0*(1.0D0+Y*Y1)*(1.0D0+Z*Z1)*(-2.0D0*X-27.0D0  FMK00380
     10*X*X*X1+9.0D0*X1)/64.0D0                                            FMK00390
      DY(X,  Z,X1,Y1,Z1)=9.0D0*(1.0D0-X*X)*(1.0D0+9.0D0*X*X1)*Y1*(1.0D0+    FMK00400
     1Z*Z1)/64.0D0                                                         FMK00410
      DZ(X,Y,  X1,Y1,Z1)=DY(X,  Y,X1,Z1,Y1)                                FMK00420
C                                                                          FMK00430
C*******************************************************************************
```

```
C                                                                       FMK00440
C           COMPUTATION OF DERIVATIVES OF SHAPE FUNCTIONS WITH RESPECT  FMK00450
C           TO 'XI','ETA','ZETA' EVALUATED AT THE POINTS OF INTEGRATION FMK00460
C                                                                       FMK00490
C             (1) CORNER NODES :    XI= +1 OR -1                        FMK00500
C                                  ZETA= +1 OR -1                       FMK00510
C                                   ETA= +1 OR -1                       FMK00520
C                          NODE # : 1,4,7,10,21,24,27,30                FMK00530
C                                                                       FMK00540
      DO 100 I=1,2                                                      FMK00550
      II=20*(I-1)+1                                                     FMK00560
      IT=II+9                                                           FMK00570
      DO 100 J=II,IT,3                                                  FMK00580
      X1=CORDG(J,1)                                                     FMK00590
      Y1=CORDG(J,2)                                                     FMK00600
      Z1=CORDG(J,3)                                                     FMK00610
      W1(1,J)=DFX(X,Y,Z,X1,Y1,Z1)                                       FMK00620
      W1(2,J)=DFX(Y,X,Z,Y1,X1,Z1)                                       FMK00630
  100 W1(3,J)=DFX(Z,X,Y,Z1,X1,Y1)                                       FMK00640
C                                                                       FMK00650
C             (2) MIDDLE NODES :  XI= + OR - 1/3                        FMK00660
C                          NODE # : 2,3,8,9,22,23,28,29                 FMK00670
C                                                                       FMK00680
      DO 200 K=1,2                                                      FMK00690
      IK=20*(K-1)                                                       FMK00700
      DO 200 I=1,2                                                      FMK00710
      II=6*(I-1)+2+IK                                                   FMK00720
      IT=II+1                                                           FMK00730
      DO 200 J=II,IT                                                    FMK00740
      X1=CORDG(J,1)                                                     FMK00750
      Y1=CORDG(J,2)                                                     FMK00760
      Z1=CORDG(J,3)                                                     FMK00770
      W1(1,J)=DX(X,Y,Z,X1,Y1,Z1)                                        FMK00780
      W1(2,J)=DY(X,  Z,X1,Y1,Z1)                                        FMK00790
  200 W1(3,J)=DZ(X,Y,  X1,Y1,Z1)                                        FMK00800
C                                                                       FMK00810
C             (3) MIDDLE NODES :  ETA= + OR - 1/3                       FMK00820
C                          NODE # : 5,6,11,12,25,26,31,32               FMK00830
C                                                                       FMK00840
      DO 300 K=1,2                                                      FMK00850
      IK=20*(K-1)                                                       FMK00860
      DO 300 I=1,2                                                      FMK00870
      II=6*(I-1)+5+IK                                                   FMK00880
      IT=II+1                                                           FMK00890
      DO 300 J=II,IT                                                    FMK00900
      X1=CORDG(J,1)                                                     FMK00910
      Y1=CORDG(J,2)                                                     FMK00920
      Z1=CORDG(J,3)                                                     FMK00930
```

```
      W1(1,J)=DY(Y,   Z,Y1,X1,Z1)                                       FMK00940
      W1(2,J)=DX(Y,X,Z,Y1,X1,Z1)                                        FMK00950
  300 W1(3,J)=DZ(Y,X,   Y1,X1,Z1)                                       FMK00960
C                                                                       FMK00970
CCC               (4) MIDDLE NODES :  ZETA= + OR - 1/3                  FMK00980
CCC                          NODE # : 13,14,15,16,17,18,19,20           FMK00990
C                                                                       FMK01000
      DO 400 K=1,2                                                      FMK01010
      IK=4*(K-1)                                                        FMK01020
      II=13+IK                                                          FMK01030
      IT=II+3                                                           FMK01040
      DO 400 J=II,IT                                                    FMK01050
      X1=CORDG(J,1)                                                     FMK01060
      Y1=CORDG(J,2)                                                     FMK01070
      Z1=CORDG(J,3)                                                     FMK01080
      W1(1,J)=DZ(Z,Y,   Z1,Y1,X1)                                       FMK01090
      W1(2,J)=DY(Z,   X,Z1,Y1,X1)                                       FMK01100
  400 W1(3,J)=DX(Z,Y,X,Z1,Y1,X1)                                        FMK01110
C                     *                       *
CCC                   *                       *
CCCCC                 *                       *
CCCCCCC       SAME AS FOR QUADRATIC ELEMENTS
CCCCC                 *                       *
CCC                   *                       *
C                     *                       *
      RETURN                                                            FMK01890
      END                                                               FMK01900
C
CCC*****************************************************************************
CCC*****************************************************************************
C


      SUBROUTINE MERGE                                                  MRG00020
C                                                                       MRG00030
      IMPLICIT REAL*8(A-H,O-Z)                                          MRG00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  MRG00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL             MRG00060
      COMMON /NB2/ NCON(32),NBC(4)                                      MRG00070
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),     MRG00080
     1RB3(96)                                                           MRG00090
      DIMENSION A(96,96),S(96,96),B(9216),T(9216),LM(32)                MRG00100
C                     *                       *
CCC                   *                       *
CCCCC                 *                       *
CCCCCC        SAME AS FOR QUADRATIC ELEMENTS
C                     *                       *
```

```
C                    *                    *
C                    *                    *
      RETURN                                                          MRG01080
      END                                                             MRG01090
C
CC*****************************************************************************
CC*****************************************************************************
C

      SUBROUTINE FLOAD                                                FLD00020
C                                                                     FLD00030
      IMPLICIT REAL*8(A-H,O-Z)                                        FLD00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,NM,NS,NCOUNT,NST,NSTF,NCLD,FLD00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDF2,NLINE,NREC7,KEL           FLD00060
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                        FLD00070
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),   FLD00080
     1RB3(96)                                                         FLD00090
      DIMENSION   A(96,96),S(96,96),R1(96),R2(96)                     FLD00100
C                    *                    *
CC                   *                    *
CCC                  *                    *
CCCC          SAME AS FOR QUADRATIC ELEMENTS
CCCCC                *                    *
CCCCC                *                    *
CCCCC                *                    *
      RETURN                                                          FLD00570
      END                                                             FLD00580
C
CC*****************************************************************************
CC*****************************************************************************
C

      SUBROUTINE BCOND                                                BC000020
C                                                                     BC000030
      IMPLICIT REAL*8(A-H,O-Z)                                        BC000040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,NM,NS,NCOUNT,NST,NSTF,NCLD,BC000050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDF2,NLINE,NREC7,KEL           BC000060
      COMMON /NB2/ NCON(32),NBC(4)                                    BC000070
      COMMON /B1/ ELDAT(10,3),CLOAD(4),BCON(3)                        BC000080
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),   BC000090
     1RB3(96)                                                         BC000100
      DIMENSION AKW1(9216),AKW2(9216)                                 BC000110
C                    *                    *
CC                   *                    *
```

```
C                    *                    *
C              SAME AS FOR QUADRATIC ELEMENTS
C                    *                    *
C                    *                    *
C                    *                    *
      RETURN                                                                BC003530
      END                                                                   BC003540
C
C
C*********************************************************************************
C*********************************************************************************
C

      SUBROUTINE SOLVE                                                      SLV00020
C                                                                           SLV00030
      IMPLICIT REAL*8(A-H,O-Z)                                              SLV00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,      SLV00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 SLV00060
      COMMON /B3/ AT1(96,96),AT2(96,96),AT3(96,96),RT1(96),RT2(96),         SLV00070
     1RT3(96)                                                               SLV00080
      DIMENSION AK1(9216),AK2(9216),AK3(9216),RB1(96),RB2(96),RB3(96)       SLV00090
C                    *                    *
C                    *                    *
C                    *                    *
C              SAME AS FOR QUADRATIC ELEMENTS
C                    *                    *
C                    *                    *
C                    *                    *
  500 RETURN                                                                SLV02260
      END                                                                   SLV02270
C
C
C*********************************************************************************
C*********************************************************************************
C

      SUBROUTINE DISP                                                       DSP00020
C                                                                           DSP00030
      IMPLICIT REAL*8(A-H,O-Z)                                              DSP00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,      DSP00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL                 DSP00060
      COMMON /NB2/ NCON(32),NBC(4)                                          DSP00070
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),         DSP00080
     1RB3(96)                                                               DSP00090
C                    *                    *
C                    *                    *
```

```
C                *                              *
C          SAME AS FOR QUADRATIC ELEMENTS
CCC              *                              *
CCC              *                              *
C                *                              *
      END                                                                DSP01320
C
C*****************************************************************************
C*****************************************************************************
C

      SUBROUTINE STRESS(I)                                               STR00020
C                                                                        STR00030
      IMPLICIT REAL*8(A-H,O-Z)                                           STR00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,   STR00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDF2,NLINE,NREC7,KEL              STR00060
      COMMON /NB2/ NCON(32),NBC(4)                                       STR00070
      COMMON /B2/ ELAST(6,6),COORD(3),CORE(32,3),UJNT(3),SSSS1(7),       STR00080
     1SSNN1(7)                                                           STR00090
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),      STR00100
     1RB3(96)                                                            STR00110
      DIMENSION SNELM(32,6),UEL(96),SSELM(32,6),SSNN(6),B(6,96),SSSS(6),STR00120
     1CORDG(32,3),LM(32)                                                 STR00130
      EQUIVALENCE (AK3(1,1),B(1)),(AK3(1,13),SNELM(1)),(AK3(1,15),       STR00140
     1SSELM(1)),(AK3(1,17),SSNN(1)),(AK3(1,17),SSSS(1)),                 STR00150
     2(UEL(1),RB1(1))                                                    STR00160
      DATA CORDG/1.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,    STR00170
     1-1.0D0,-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,STR00180
     21.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,  STR00190
     31.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,-1.0D0,-1.0D0,STR00200
     4-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,1.0D0,1.0D0,  STR00210
     51.0D0,1.0D0,1.0D0,1.0D0,0.3333333333333333,-0.3333333333333333,   STR00220
     6-1.0D0,-1.0D0,-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333STR00230
     7,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0),-1.0D0,-1.0D0,1.0D0,1.0D0,STR00240
     81.0D0,1.0D0,0.3333333333333333,-0.3333333333333333,-1.0D0,-1.0D0, STR00250
     9-1.0D0,-1.0D0,-0.3333333333333333,0.3333333333333333,1.0D0,1.0D0, STR00260
     A1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,      STR00270
     B0.3333333333333333,0.3333333333333333,0.3333333333333333,0.3333333STR00280
     C333333333,-0.3333333333333333,-0.3333333333333333 ,-0.333333333333STR00290
     D3333,-0.3333333333333333,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0STR00300
     E,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0/                       STR00310
C                *                              *
CCC              *                              *
CCC              *                              *
C          SAME AS FOR QUADRATIC ELEMENTS
C                *                              *
```

```
C                     *                     *
C                     *                     *
      END                                                                STR01080
C
C
C*****************************************************************************
C*****************************************************************************
C


      SUBROUTINE DISK                                                    DSK00020
                                                                         DSK00030
C                                                                        DSK00040
      IMPLICIT REAL*8(A-H,O-Z)                                           DSK00050
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,   DSK00060
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL              DSK00070
      COMMON /B2/ ELAST(6,6),COORD(3),CORE(32,3),UJNT(3),SSSS1(7),       DSK00080
     1SSNN1(7)                                                           DSK00090
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),      DSK00100
     1RB3(96)                                                            DSK00110
      DIMENSION STK1(768),STK2(768),STK3(768),STK4(768),STK5(768),       DSK00120
     2          STK6(768),STK7(768),STK8(768),STK9(768),STK10(768),      DSK00130
     3          STK11(768),STK12(768),A(1)                               DSK00140
      EQUIVALENCE (STK1(1),AK1(1,1)),(STK2(1),AK1(1,9)),(STK3(1),AK1(1,  DSK00150
     217)),(STK4(1),AK1(1,25)),(STK5(1),AK1(1,33)),(STK6(1),AK1(1,41)),  DSK00160
     3(STK7(1),AK1(1,49)),(STK8(1),AK1(1,57)),(STK9(1),AK1(1,65)),       DSK00170
     4(STK10(1),AK1(1,73)),(STK11(1),AK1(1,81)),(STK12(1),AK1(1,89))     DSK00180
C                                                                        DSK00230
C                                                                        DSK00240
      NCOUNT=NS*NS                                                       DSK00250
      LREC7=NCOUNT/NREC7                                                 DSK00260
      NS1=NS/NDF                                                         DSK00270
      RETURN                                                             DSK00280
C                                                                        DSK00290
C*****************************************************************************DSK00300
C                                                                        DSK00310
      ENTRY WRDISK(NTRACK,A)                                             DSK00320
C                                                                        DSK00330
      I7=(NTRACK-1)*NREC7+1                                              DSK00340
      N1=1                                                               DSK00350
      DO 10 L=1,NREC7                                                    DSK00360
      N2=L*LREC7                                                         DSK00370
      WRITE(7'I7)            (A(J),J=N1,N2)                              DSK00380
   10 N1=N2+1                                                            DSK00390
      RETURN                                                             DSK00400
C                                                                        DSK00410
C*****************************************************************************DSK00420
C                                                                        DSK00430
      ENTRY RDDISK(NTRACK,A)
```

```
C                                                                       DSK00440
      I7=(NTRACK-1)*NREC7+1                                             DSK00450
      N1=1                                                              DSK00460
      DO 20 L=1,NREC7                                                   DSK00470
      N2=L*LREC7                                                        DSK00480
      READ(7'I7,ERR=500)            (A(J),J=N1,N2)                      DSK00490
   20 N1=N2+1                                                           DSK00500
      RETURN                                                            DSK00510
C                                                                       DSK00520
C*****************************************************************    DSK00530
C                                                                       DSK00540
      ENTRY  WDISK1(NTRACK)                                             DSK00550
C                                                                       DSK00560
      I8=(NTRACK-1)*12+1                                                DSK00570
      WRITE(8'I8) STK1                                                  DSK00580
      WRITE(8'I8) STK2                                                  DSK00590
      WRITE(8'I8) STK3                                                  DSK00600
      WRITE(8'I8) STK4                                                  DSK00610
      WRITE(8'I8) STK5                                                  DSK00620
      WRITE(8'I8) STK6                                                  DSK00630
      WRITE(8'I8) STK7                                                  DSK00640
      WRITE(8'I8) STK8                                                  DSK00650
      WRITE(8'I8) STK9                                                  DSK00660
      WRITE(8'I8) STK10                                                 DSK00670
      WRITE(8'I8) STK11                                                 DSK00680
      WRITE(8'I8) STK12                                                 DSK00690
      RETURN                                                            DSK00700
C                                                                       DSK00710
C*****************************************************************    DSK00720
C                                                                       DSK00730
      ENTRY  RDISK1(NTRACK,A)                                           DSK00740
C                                                                       DSK00750
      I8=(NTRACK-1)*12+1                                                DSK00760
      READ(8'I8,ERR=500) STK1                                           DSK00770
      READ(8'I8,ERR=500) STK2                                           DSK00780
      READ(8'I8,ERR=500) STK3                                           DSK00790
      READ(8'I8,ERR=500) STK4                                           DSK00800
      READ(8'I8,ERR=500) STK5                                           DSK00810
      READ(8'I8,ERR=500) STK6                                           DSK00820
      READ(8'I8,ERR=500) STK7                                           DSK00830
      READ(8'I8,ERR=500) STK8                                           DSK00840
      READ(8'I8,ERR=500) STK9                                           DSK00850
      READ(8'I8,ERR=500) STK10                                          DSK00860
      READ(8'I8,ERR=500) STK11                                          DSK00870
      READ(8'I8,ERR=500) STK12                                          DSK00880
      RETURN                                                            DSK00890
C                     *                       *
C                     *                       *
```

```
C                    *                    *
C              SAME AS FOR QUADRATIC ELEMENTS
C                    *                    *
C                    *                    *
C                    *                    *
      STOP                                                          DSK01990
      END                                                           DSK02000
C
C
C*****************************************************************************
C*****************************************************************************
C

      SUBROUTINE ERROR                                              ERR00020
C                    *                    *
C                    *                    *
C                    *                    *
C              SAME AS FOR QUADRATIC ELEMENTS
C                    *                    *
C                    *                    *
C                    *                    *
      STOP                                                          ERR04780
      END                                                           ERR04790
C
C
C*****************************************************************************
C*****************************************************************************
C

      SUBROUTINE SKIP                                               SKP00020
C                    *                    *
C                    *                    *
C                    *                    *
C              SAME AS FOR QUADRATIC ELEMENTS
C                    *                    *
C                    *                    *
C                    *                    *
      RETURN                                                        SKP00170
      END                                                           SKP00180
C
C
C*****************************************************************************
C*****************************************************************************
C
```

```
      SUBROUTINE SORT                                                    SRT00020
C                                                                        SRT00030
      IMPLICIT REAL*8(A-H,O-Z)                                           SRT00040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,   SRT00050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL              SRT00060
      COMMON /NB2/ NCON(32),NBC(4)                                       SRT00070
      COMMON /B2/ ELAST(6,6),COORD(3),COREL(32,3),UJNT(3),SSSS1(7),      SRT00080
     1SSNN1(7)                                                           SRT00090
C                     *                  *
C                     *                  *
C                     *                  *
C            SAME AS FOR QUADRATIC ELEMENTS
C                     *                  *
C                     *                  *
C                     *                  *
      RETURN                                                             SRT00270
      END                                                                SRT00280
C
C*************************************************************************
C*************************************************************************
C

      SUBROUTINE MULT(A,B,C,NRA,NCA,NCB)                                 MLT00020
C                     *                  *
C                     *                  *
C                     *                  *
C            SAME AS FOR QUADRATIC ELEMENTS
C                     *                  *
C                     *                  *
C                     *                  *
      RETURN                                                             MLT00150
      END                                                                MLT00160
C
C*************************************************************************
C*************************************************************************
C

      SUBROUTINE SYMINV(A,N,B,C,IFLG)                                    SMV00020
C                     *                  *
C                     *                  *
C                     *                  *
C            SAME AS FOR QUADRATIC ELEMENTS
C                     *                  *
C                     *                  *
```

```
C                    *                    *                    SMV01010
                                                                SMV01020
      RETURN
      END
```

```
      REAL*8 ALPHA                                                      JCL00010
      DIMENSION NRD1(3),NRD2(5),NRS(6),LRWD1(2),LRWD2(5),NRA(4),LRBA(4),JCL00020
     1NCON(32),FMT(20),LRBD1(3),LRBD2(5)                                JCL00030
      INTEGER LRBS(6)/32,16,4,32,480,24/,NFD1(3)/7,8,9/,NFD2(5)/10,12,13JCL00040
     A,20,21/,NFS(6)/14,15,16,17,19,22/                                 JCL00050
      DATA IBL/'    '/,IZR/'0   '/                                      JCL00060
C                                                                       JCL00070
CC*****************************************************************  JCL00080
CC                                                                      JCL00090
CCC            GENEPAL CHARACTERISTICS OF PROBLEM                       JCL00100
C                                                                       JCL00110
    2 READ(5,1000) NEL,NPEL,MM,NS,NDPT,NPBC,NPBC2,NCLD,NREC7,NMAT       JCL00120
      IF(NEL.LT.0) STOP                                                 JCL00130
      NDF=3                                                             JCL00140
      NCOUNT=NS*NS                                                      JCL00150
      NST=NPEL*NDF                                                      JCL00160
      NSTF=NST*NST                                                      JCL00170
      NEQ=NDPT*NDF                                                      JCL00180
      NN=(NEQ+NS-1)/NS                                                  JCL00190
      IF(MM.NE.0) GO TO 10                                              JCL00200
      NBAND=0                                                           JCL00210
      READ(5,1300) FMT                                                  JCL00220
      DO 5 I=1,NEL                                                      JCL00230
      READ(5,FMT) (NCON(J),J=1,NPEL)                                    JCL00240
      NPELM=NPEL-1                                                      JCL00250
      DO 5 K=1,NPELM                                                    JCL00260
      JP=K+1                                                            JCL00270
      DO 5 L=JP,NPEL                                                    JCL00280
    5 NBAND=MAXO(NBAND,IABS(NCON(K)-NCON(L)))                           JCL00290
      NBAND=(NBAND+1)*NDF                                               JCL00300
      MM=(NBAND+2*(NS-1))/NS                                            JCL00310
      IF(NN.LT.MM) MM=NN                                                JCL00320
   10 WRITE(6,9500)                                                     JCL00330
      WRITE(6,7500)                                                     JCL00340
      WRITE(6,1500) NEL,NDPT,NMAT,NS,NPBC,NPBC2,NCLD,NN,MM,NPEL,NREC7   JCL00350
      WRITE(6,9500)                                                     JCL00360
C                                                                       JCL00370
CC*****************************************************************  JCL00380
CC                                                                      JCL00390
CCCC           INITIALIZATION OF VARIABLE FILE DATA                     JCL00400
CCCCCC                                                                  JCL00410
CCCCCC           (1) DIRECT ACCESS FILES                                JCL00420
CCCCCC                                                                  JCL00430
CCCCCC               FILE NO 7                                          JCL00440
C                                                                       JCL00450
      NRD1(1)=MM*NN*NREC7                                               JCL00460
      LRWD1(1)=NCOUNT*2/NREC7                                           JCL00470
      LRBD1(1)=LRWD1(1)*4                                               JCL00480
```

```
C                                                                       JCL00490
CC                    FILE NO 8                                         JCL00500
C                                                                       JCL00510
      NREC8=4                                                           JCL00520
      IF(NPEL.EQ.32) NREC8=12                                           JCL00530
      NRD1(2)=NEL*NREC8                                                 JCL00540
      LRWD1(2)=NSTF*2/NREC8                                             JCL00550
      LRBD1(2)=LRWD1(2)*4                                               JCL00560
C                                                                       JCL00570
CC                    FILE NO 10                                        JCL00580
C                                                                       JCL00590
      NRD2(1)=NDPT                                                      JCL00600
      LRWD2(1)=6                                                        JCL00610
      LRBD2(1)=LRWD2(1)*4                                               JCL00620
C                                                                       JCL00630
CC                    FILE NO 12                                        JCL00640
C                                                                       JCL00650
      NRD2(2)=NDPT                                                      JCL00660
      LRWD2(2)=14                                                       JCL00670
      LRBD2(2)=LRWD2(2)*4                                               JCL00680
C                                                                       JCL00690
CC                    FILE NO 13                                        JCL00700
C                                                                       JCL00710
      NRD2(3)=NDPT                                                      JCL00720
      LRWD2(3)=14                                                       JCL00730
      LRBD2(3)=LRWD2(3)*4                                               JCL00740
C                                                                       JCL00750
CC                    FILE NO 20                                        JCL00760
C                                                                       JCL00770
      NRD2(4)=NN                                                        JCL00780
      LRWD2(4)=NS*2                                                     JCL00790
      LRBD2(4)=LRWD2(4)*4                                               JCL00800
C                                                                       JCL00810
CC                    FILE NO 21                                        JCL00820
C                                                                       JCL00830
      NRD2(5)=NN*NS/NDF                                                 JCL00840
      LRWD2(5)=6                                                        JCL00850
      LRBD2(5)=24                                                       JCL00860
C                                                                       JCL00870
CC                (2) SEQUENTIAL FILES                                  JCL00880
C                                                                       JCL00890
C                                                                       JCL00900
CC                    FILE NO 9                                         JCL00910
C                                                                       JCL00920
      NRD1(3)=NEL                                                       JCL00930
      LRBD1(3)=128                                                      JCL00940
      IF(NPEL.EQ.20) LRBD1(3)=80                                        JCL00950
C                                                                       JCL00960
```

```
C                  FILES NO 14,15,16,17,19,22                                  JCL00970
C                                                                              JCL00980
      NRS(1)=NCLD                                                              JCL00990
      NRS(2)=NPBC                                                              JCL01000
      NRS(3)=NEL                                                               JCL01010
      NRS(4)=NPBC                                                              JCL01020
      NRS(5)=NEL                                                               JCL01030
      NRS(6)=NPBC2                                                             JCL01040
      IF(NMAT.GT.1) GO TO 12                                                   JCL01050
      NRS(3)=0                                                                 JCL01060
      LRBS(3)=0                                                                JCL01070
   12 IF(NPBC2.GT.0) GO TO 15                                                  JCL01080
      NRS(6)=0                                                                 JCL01090
      LRBS(6)=0                                                                JCL01100
C                                                                              JCL01110
C*****************************************************************************JCL01120
C                                                                              JCL01130
CC             CONTROL CARDS                                                   JCL01140
C                                                                              JCL01150
   15 WRITE(6,8000)                                                            JCL01160
      WRITE(7,8000)                                                            JCL01170
      WRITE(6,9600)                                                            JCL01180
C                                                                              JCL01190
CC                FILES NO 7,8,9                                               JCL01200
C                                                                              JCL01210
      DO 30 I=1,3                                                              JCL01220
      NR=NRD1(I)                                                               JCL01230
      LRB=LRBD1(I)                                                             JCL01240
      CALL EXDIC(NR,4,0,0,IER,ANR,ALPHA)                                       JCL01250
      CALL EXDIC(LRB,4,0,0,IER,ALRB,ALPHA)                                     JCL01260
      REWIND 8                                                                 JCL01270
      WRITE(8,2000) ANR                                                        JCL01280
      WRITE(8,2000) ALRB                                                       JCL01290
      REWIND 8                                                                 JCL01300
      READ(8,3000) NRA                                                         JCL01310
      READ(8,3000) LRBA                                                        JCL01320
      DO 20 J=1,4                                                              JCL01330
      IF(NRA(J).EQ.IBL) NRA(J)=IZR                                             JCL01340
      IF(LRBA(J).EQ.IBL) LRBA(J)=IZR                                           JCL01350
   20 CONTINUE                                                                 JCL01360
      WRITE(6,4000) NFD1(I),LRBA,NRA                                           JCL01370
      WRITE(7,4100) NFD1(I),LRBA,NRA                                           JCL01380
   30 CONTINUE                                                                 JCL01390
C                                                                              JCL01400
CC                FILES NO 10,12,13,20,21                                      JCL01410
C                                                                              JCL01420
      DO 50 I=1,5                                                              JCL01430
      NR=NRD2(I)                                                               JCL01440
```

```
      LRB=LRBD2(I)                                                      JCL01450
      CALL EXDIC(NR,4,0,0,IER,ANR,ALPHA)                                JCL01460
      CALL EXDIC(LRB,4,0,0,IER,ALRB,ALPHA)                              JCL01470
      REWIND 8                                                          JCL01480
      WRITE(8,2000) ANR                                                 JCL01490
      WRITE(8,2000) ALRB                                                JCL01500
      REWIND 8                                                          JCL01510
      READ(8,3000) NRA                                                  JCL01520
      READ(8,3000) LRBA                                                 JCL01530
      DO 40 J=1,4                                                       JCL01540
      IF(NRA(J).EQ.IBL) NRA(J)=IZR                                      JCL01550
      IF(LRBA(J).EQ.IBL) LRBA(J)=IZR                                    JCL01560
   40 CONTINUE                                                          JCL01570
      WRITE(6,5000) NFD2(I),LRBA,NRA                                    JCL01580
      WRITE(7,5100) NFD2(I),LRBA,NRA                                    JCL01590
   50 CONTINUE                                                          JCL01600
C                                                                       JCL01610
CC                FILES NO 14,15,16,17,22                               JCL01620
C                                                                       JCL01630
      DO 70 I=1,6                                                       JCL01640
      IF((I.EQ.3).AND.(NMAT.EQ.1)) GO TO 70                             JCL01650
      IF((I.EQ.6).AND.(NPBC2.EQ.0)) GO TO 70                            JCL01660
      NR=NRS(I)                                                         JCL01670
      LRB=LRBS(I)                                                       JCL01680
      CALL EXDIC(NR,4,0,0,IER,ANR,ALPHA)                                JCL01690
      CALL EXDIC(LRB,4,0,0,IER,ALRB,ALPHA)                              JCL01700
      REWIND 8                                                          JCL01710
      WRITE(8,2000) ANR                                                 JCL01720
      WRITE(8,2000) ALRB                                                JCL01730
      REWIND 8                                                          JCL01740
      READ(8,3000) NRA                                                  JCL01750
      READ(8,3000) LRBA                                                 JCL01760
      DO 60 J=1,4                                                       JCL01770
      IF(NRA(J).EQ.IBL) NRA(J)=IZR                                      JCL01780
      IF(LRBA(J).EQ.IBL) LRBA(J)=IZR                                    JCL01790
   60 CONTINUE                                                          JCL01800
      WRITE(6,5000) NFS(I),LRBA,NRA                                     JCL01810
      WRITE(7,5100) NFS(I),LRBA,NRA                                     JCL01820
   70 CONTINUE                                                          JCL01830
C                                                                       JCL01840
C**********************************************************************JCL01850
CC                                                                      JCL01860
CC                DEFINE FILE STATEMENTS                                JCL01870
C                                                                       JCL01880
  120 WRITE(6,9500)                                                     JCL01890
      WRITE(6,9000)                                                     JCL01900
      WRITE(7,9000)                                                     JCL01910
      WRITE(6,9600)                                                     JCL01920
```

```
C                                                                               JCL01930
      IC2=2                                                                     JCL01940
      IC3=3                                                                     JCL01950
      ICD1=150                                                                  JCL01960
      ICD2=160                                                                  JCL01970
      ICD3=170                                                                  JCL01980
      ICD4=180                                                                  JCL01990
      IF(NPEL.EQ.20) GO TO 130                                                  JCL02000
      ICD1=190                                                                  JCL02010
      ICD2=200                                                                  JCL02020
      ICD3=210                                                                  JCL02030
      ICD4=220                                                                  JCL02040
  130 WRITE(6,6000) NFD1(1),NRD1(1),LRWD1(1),NFD1(1),NFD1(2),NRD1(2),           JCL02050
     ALRWD1(2),NFD1(2),ICD1,NFD2(1),NRD2(1),LRWD2(1),NFD2(1),ICD2               JCL02060
      WRITE(6,7000) IC2,NFD2(2),NRD2(2),LRWD2(2),NFD2(2),NFD2(3),NRD2(3)JCL02070
     A,LRWD2(3),NFD2(3),ICD3                                                    JCL02080
      WRITE(6,7050) IC3,NFD2(4),NPD2(4),LRWD2(4),NFD2(4),NFD2(5),NRD2(5)JCL02090
     A,LRWD2(5),NFD2(5),ICD4                                                    JCL02100
      WRITE(7,6100) NFD1(1),NRD1(1),LRWD1(1),NFD1(1),NFD1(2),NRD1(2),           JCL02110
     ALRWD1(2),NFD1(2),ICD1,NFD2(1),NRD2(1),LRWD2(1),NFD2(1),ICD2               JCL02120
      WRITE(7,7100) IC2,NFD2(2),NRD2(2),LRWD2(2),NFD2(2),NFD2(3),NRD2(3)JCL02130
     A,LRWD2(3),NFD2(3),ICD3                                                    JCL02140
      WRITE(7,7200) IC3,NFD2(4),NRD2(4),LRWD2(4),NFD2(4),NFD2(5),NRD2(5)JCL02150
     A,LRWD2(5),NFD2(5),ICD4                                                    JCL02160
C                                                                               JCL02170
C*****************************************************************************JCL02180
C                                                                               JCL02190
C             GENERAL FILE DATA                                                 JCL02200
C                                                                               JCL02210
      WRITE(6,9500)                                                             JCL02220
      WRITE(6,9800)                                                             JCL02230
      WRITE(6,9600)                                                             JCL02240
      WRITE(7,9800)                                                             JCL02250
      WRITE(6,9910)                                                             JCL02260
      WRITE(6,9900) (NRD1(I),I=1,3),(NRD2(I),I=1,2),(LRBD1(I),I=1,3),           JCL02270
     1(LRBD2(I),I=1,2),(LRWD1(I),I=1,2),(LRWD2(I),I=1,2),NREC7,                 JCL02280
     2NRD2(3),(NRS(I),I=1,4),LRBD2(3),(LRBS(I),I=1,4),LRWD2(3),NRS(5),          JCL02290
     3(NRD2(I),I=4,5),NRS(6),LRBS(5),(LRBD2(I),I=4,5),LRBS(6),                  JCL02300
     4(LRWD2(I),I=4,5)                                                          JCL02310
      WRITE(6,9920)                                                             JCL02320
      WRITE(7,3500) NRD1(1),LRBD1(1),LRWD1(1),NREC7,NRD1(2),LRBD1(2),           JCL02330
     1LRWD1(2),NRD1(3),LRBD1(3),                                                JCL02340
     2(NRD2(I),LRBD2(I),LRWD2(I),I=1,3),(NRS(I),LRBS(I),I=1,5),(NRD2(I),JCL02350
     3LRBD2(I),LRWD2(I),I=4,5),NRS(6),LRBS(6)                                   JCL02360
C                                                                               JCL02370
      GO TO 2                                                                   JCL02380
C                                                                               JCL02390
C*****************************************************************************JCL02400
```

```
C                                                                               JCL02410
 1000 FORMAT(1X,10I5)                                                           JCL02420
 1300 FORMAT(20A4)                                                              JCL02430
 1500 FORMAT(//,                                                                JCL02440
     15X,'NEL....TOTAL NUMBER OF ELEMENTS.....................',I4/,          JCL02450
     25X,'NDPT...TOTAL NUMBER OF NODAL POINTS.................',I4/,          JCL02460
     35X,'NMAT...NUMBER OF DIFFERENT MATERIALS................',I4/,          JCL02470
     45X,'       (MAXIMUM IS  10)',/,                                         JCL02480
     55X,'NS.....BLOCK SIZE FOR THE LARGE CAPACITY SOLVER.....',I4/,          JCL02490
     65X,'NPBC...NUMBER OF NODAL POINTS WITH BOUND. COND......',I4/,          JCL02500
     75X,'NPBC2..NUMBER OF NODAL POINTS WITH PREDEF. DISPL....',I4/,          JCL02510
     85X,'NCLD...NUMBER OF NODAL POINTS WITH CONC. LOAD.......',I4/,          JCL02520
     95X,'NN.....NUMBER OF BLOCKS PER COLUMN..................',I4/,          JCL02530
     A5X,'MM.....NUMBER OF BLOCKS PER ROW.....................',I4/,          JCL02540
     B5X,'NPEL...NUMBER OF NODES PER ELEMENT..................',I4/,          JCL02550
     C5X,'NREC7..NUMBER OF RECORDS OF FILE NO 7...............',I4/)          JCL02560
 2000 FORMAT(A4)                                                                JCL02570
 3000 FORMAT(4A1)                                                               JCL02580
 3500 FORMAT(7I10)                                                              JCL02590
 4000 FORMAT(' ',4X,'//GO.FT0',I1,'F001 DD UNIT=SYSDA,SPACE=(',4A1,',(',       JCL02600
     24A1,',1)),DISP=(NEW,DELETE)')                                             JCL02610
 4100 FORMAT('//GO.FT0',I1,'F001 DD UNIT=SYSDA,SPACE=(',4A1,',(',4A1,          JCL02620
     2',1)),DISP=(NEW,DELETE)')                                                 JCL02630
 5000 FORMAT(' ',4X,'//GO.FT',I2,'F001 DD UNIT=SYSDA,SPACE=(',4A1,',(',        JCL02640
     24A1,',1)),DISP=(NEW,DELETE)')                                             JCL02650
 5100 FORMAT('//GO.FT',I2,'F001 DD UNIT=SYSDA,SPACE=(',4A1,',(',4A1,           JCL02660
     2',1)),DISP=(NEW,DELETE)')                                                 JCL02670
 6000 FORMAT(' ',10X,'DEFINE FILE',I2,'(',I4,',',I4,',U,I',I1,'),',I1,'(JCL02680
     A',I4,',',I4,',U,I',I1,'),',18X,'DSK00',I3/' ',9X,'1',11X,I2,'(',I4JCL02690
     B,',',I4,',U,I',I2,'),',35X,'DSK00',I3)                                    JCL02700
 6100 FORMAT(6X,'DEFINE FILE',I2,'(',I4,',',I4,',U,I',I1,'),',I1,'(',          JCL02710
     AI4,',',I4,',U,I',I1,'),',18X,'DSK00',I3/5X,'1',11X,I2,'(',I4,',',IJCL02720
     B4,',U,I',I2,'),',35X,'DSK00',I3)                                          JCL02730
 7000 FORMAT(' ',9X,I1,10X,I3,'(',I4,',',I4,',U,I',I2,'),',I2,'(',I4,          JCL02740
     A',',I4,',U,I',I2,'),',15X,'DSK00',I3)                                     JCL02750
 7050 FORMAT(' ',9X,I1,10X,I3,'(',I4,',',I4,',U,I',I2,'),',I2,'(',I4,          JCL02760
     A',',I4,',U,I',I2,')',16X,'DSK00',I3)                                      JCL02770
 7100 FORMAT(5X,I1,10X,I3,'(',I4,',',I4,',U,I',I2,'),',I2;'(',I4,              JCL02780
     A',',I4,',U,I',I2,'),',15X,'DSK00',I3)                                     JCL02790
 7200 FORMAT(5X,I1,10X,I3,'(',I4,',',I4,',U,I',I2,'),',I2,'(',I4,              JCL02800
     A',',I4,',U,I',I2,')',16X,'DSK00',I3)                                      JCL02810
 7500 FORMAT('1',18X,'****** INPUT DATA ******'/)                               JCL02820
 8000 FORMAT(' ',10X,'****** JCL CARDS ******')                                 JCL02830
 9000 FORMAT(' ',15X,'****** DEFINE FILE STATEMENTS ******')                    JCL02840
 9500 FORMAT(////)                                                              JCL02850
 9600 FORMAT(//)                                                                JCL02860
 9800 FORMAT(' ',17X,'****** GENERAL FILE DATA ******')                         JCL02870
 9900 FORMAT(' ',5X,'NR7  =',I4,4X,'NR8  =',I4,4X,'NR9  =',I4,4X,'NR10 =JCL02880
```

```
     1',I4,4X,'NR12 =',I4/' ',                                                  JCL02890
     25X,'LRB7 =',I4,4X,'LRB8 =',I4,4X,'LRB9 =',I4,4X,'LRB10=',I4,4X,           JCL02900
     3'LRB12=',I4/' ',5X,'LRW7 =',I4,4X,'LRW8 =',I4,18X,'LRW10=',I4,4X,         JCL02910
     4'LRW12 =',I4/' ',5X,'NREC7=',I4//' ',                                     JCL02920
     55X,'NR13 =',I4,4X,'NR14 =',I4,4X,'NR15 =',I4,4X,'NR16 =',I4,4X,           JCL02930
     6'NR17 =',I4/' ',                                                          JCL02940
     75X,'LRB13=',I4,4X,'LRB14=',I4,4X,'LRB15=',I4,4X,'LRB16=',I4,4X,           JCL02950
     8'LRB17=',I4/' ',                                                          JCL02960
     95X,'LRW13=',I4,//' ',                                                     JCL02970
     A5X,'NR19 =',I4,4X,'NR20 =',I4,4X,'NR21 =',I4,4X,'NR22 =',I4/' ',         JCL02980
     B5X,'LRB19=',I4,4X,'LRB20=',I4,4X,'LRB21=',I4,4X,'LRB22=',I4/' ',         JCL02990
     C19X,'LRW20=',I4,4X,'LRW21=',I4//)                                         JCL03000
9910 FORMAT(' ',5X,'NRX  = NUMBER OF RECORDS OF FILE X'/' ',                    JCL03010
     25X,'LRBX = LENGTH OF RECORD OF FILE X IN BYTES'/' ',                      JCL03020
     35X,'LRWX =  -||-      -||-     -||-      WORDS'//)                       JCL03030
9920 FORMAT(' ',5X,'NOTE : IF DATA OF FILES NO 16 OR 22 ARE ZERO THESE          JCL03040
     1FILES ARE NOT REQUIRED')                                                  JCL03050
      STOP                                                                      JCL03060
      END                                                                       JCL03070
C
CC
CC
C
EXDIC    START                                                                  EDIC0670
         PRINT NOGEN                                                            EDIC0680
         USING EXDIC,15                                                         EDIC0690
         STM   14,12,12(13)                                                     EDIC0700
         L     2,0(0,1)                                                         EDIC0710
         L     2,0(0,2)                                                         EDIC0720
         LA    1,4(0,1)                                                         EDIC0730
         L     3,0(0,1)                                                         EDIC0740
         L     3,0(0,3)                                                         EDIC0750
         LA    1,4(0,1)                                                         EDIC0760
         L     4,0(0,1)                                                         EDIC0770
         L     4,0(0,4)                                                         EDIC0780
         LA    1,4(0,1)                                                         EDIC0790
         ST    1,ADDRESS                                                        EDIC0800
         L     5,0(0,1)                                                         EDIC0810
         L     5,0(0,5)                                                         EDIC0820
         BC    15,NEXT                                                          EDIC0830
         DS    0F                                                               EDIC0840
BLANK    DC    X'40000000'                                                      EDIC0850
ALPHA    DC    X'00F0'                                                          EDIC0860
MINUS    DC    X'60'                                                            EDIC0870
         DS    0D                                                               EDIC0880
UPLIM    DC    X'F9F9F9F9F9F9F9F9'                                              EDIC0890
LOWLM1   DC    X'60F9F9F9F9F9F9F9'                                              EDIC0900
LOWLIM   DC    X'60F9F9F9F9F9F9F9'                                              EDIC0910
```

```
SLOWLIM  DC    X'4040404060F9F9F9'                                  EDIC0920
WORD     DC    X'4040404040404040'                                  EDIC0930
IER      DC    X'00000000'                                          EDIC0940
MAX1     DC    X'05F5E0FF'                                          EDIC0950
MAX      DC    X'05F5E0FF'                                          EDIC0960
MIN1     DC    X'FF676981'                                          EDIC0970
MIN      DC    X'FF676981'                                          EDIC0980
MAX2     DC    X'0000270F'                                          EDIC0990
MIN2     DC    X'FFFFFC19'                                          EDIC1000
NEXT     LTR   6,4                                                  EDIC1010
         BZ    JTISOK                                               EDIC1020
         BCT   6,*+4                                                EDIC1030
         LTR   6,6                                                  EDIC1040
         BZ    JTISOK                                               EDIC1050
         LA    0,2                                                  EDIC1060
         ST    0,IER                                                EDIC1070
         BC    15,RETURN                                            EDIC1080
JTISOK   LA    8,4                                                  EDIC1090
         CR    3,8                                                  EDIC1100
         BNE   EIGHTCAR                                             EDIC1110
         MVC   MAX(4),MAX2                                          EDIC1120
         MVC   MIN(4),MIN2                                          EDIC1130
         MVC   LOWLIM(8),SLOWLIM                                    EDIC1140
EIGHTCAP L     0,MAX                                                EDIC1150
         LR    7,2                                                  EDIC1160
         SR    7,0                                                  EDIC1170
         BC    13,NOTTOBIG                                          EDIC1180
         MVC   WORD(8),UPLIM                                        EDIC1190
         LA    0,1                                                  EDIC1200
         ST    0,IER                                                EDIC1210
         BC    15,RETURN                                            EDIC1220
NOTTOBIG L     0,MIN                                                EDIC1230
         LP    7,2                                                  EDIC1240
         SR    7,0                                                  EDIC1250
         BC    11,INRANGE                                           EDIC1260
         MVC   WORD(8),LOWLIM                                       EDIC1270
         SR    0,0                                                  EDIC1280
         BCT   0,*+4                                                EDIC1290
         ST    0,IER                                                EDIC1300
         BC    15,RETURN                                            EDIC1310
INRANGE  MVI   FLAG+1,X'01'                                        EDIC1320
         LTR   7,2                                                  EDIC1330
         BC    11,NONNEG                                            EDIC1340
         MVI   FLAG+1,X'00'                                         EDIC1350
         SR    1,1                                                  EDIC1360
         BCT   1,*+4                                                EDIC1370
         MR    6,1                                                  EDIC1380
NONNEG   LA    8,4                                                  EDIC1390
```

```
            LA      1,WORD+7                                    EDIC1400
            LR      11,1                                        EDIC1410
            LA      0,10                                        EDIC1420
            LTR     7,7                                         EDIC1430
            BC      7,NOTZERO                                   EDIC1440
            MVC     WORD+7(1),ALPHA+1                           EDIC1450
            BC      15,RETURN                                   EDIC1460
NOTZERO     SR      6,6                                         EDIC1470
            DR      6,0                                         EDIC1480
            AH      6,ALPHA                                     EDIC1490
            STH     6,TEMP-1                                    EDIC1500
            MVC     0(1,1),TEMP                                 EDIC1510
            BCT     1,*+4                                       EDIC1520
            LTR     7,7                                         EDIC1530
            BC      7,NOTZERO                                   EDIC1540
            LH      10,FLAG                                     EDIC1550
            LTR     10,10                                       EDIC1560
            BNZ     NOSIGN                                      EDIC1570
            MVC     0(1,1),MINUS                                EDIC1580
            BCT     1,*+4                                       EDIC1590
NOSIGN      LR      12,11                                       EDIC1600
            SR      12,1                                        EDIC1610
            LR      10,3                                        EDIC1620
            SR      10,12                                       EDIC1630
            BZ      RETURN                                      EDIC1640
            LTR     4,4                                         EDIC1650
            BZ      RTJSTFD                                     EDIC1660
            LA      1,1(0,1)                                    EDIC1670
            LR      14,1                                        EDIC1680
            SR      14,10                                       EDIC1690
            LR      11,12                                       EDIC1700
SHIFTL      MVC     0(1,14),0(1)                                EDIC1710
            MVC     0(1,1),BLANK                                EDIC1720
            LA      1,1(0,1)                                    EDIC1730
            LA      14,1(0,14)                                  EDIC1740
            BCT     11,SHIFTL                                   EDIC1750
            BCT     1,*+4                                       EDIC1760
            BCT     14,*+4                                      EDIC1770
            SR      10,5                                        EDIC1780
            BM      RETURN                                      EDIC1790
            LTR     11,5                                        EDIC1800
            BNP     RETURN                                      EDIC1810
            AR      11,14                                       EDIC1820
            LR      1,14                                        EDIC1830
            LR      14,12                                       EDIC1840
SECSHFTL    MVC     0(1,11),0(1)                                EDIC1850
            MVC     0(1,1),BLANK                                EDIC1860
            BCT     11,*+4                                      EDIC1870
```

```
         BCT    1,*+4                                                  EDIC1880
         BCT    14,SECSHFTL                                            EDIC1890
         BC     15,RETURN                                              EDIC1900
RTJSTFD  LTR    11,5                                                   EDIC1910
         BNP    RETURN                                                 EDIC1920
         SR     10,11                                                  EDIC1930
         BM     RETURN                                                 EDIC1940
         LA     1,1(0,1)                                               EDIC1950
         LR     14,1                                                   EDIC1960
         SR     14,11                                                  EDIC1970
         LR     11,1                                                   EDIC1980
         LR     10,12                                                  EDIC1990
SECSHFTR MVC    0(1,14),0(11)                                          EDIC2000
         MVC    0(1,11),BLANK                                          EDIC2010
         LA     11,1(0,11)                                             EDIC2020
         LA     14,1(0,14)                                             EDIC2030
         BCT    10,SECSHFTR                                            EDIC2040
RETURN   L      1,ADDRESS                                              EDIC2050
         LA     1,4(0,1)                                               EDIC2060
         L      1,0(0,1)                                               EDIC2070
         MVC    0(4,1),IER                                             EDIC2080
         L      1,ADDRESS                                              EDIC2090
         LA     1,8(0,1)                                               EDIC2100
         L      1,0(0,1)                                               EDIC2110
         MVC    0(4,1),WORD+4                                          EDIC2120
         L      1,ADDRESS                                              EDIC2130
         LA     1,12(0,1)                                              EDIC2140
         L      1,0(0,1)                                               EDIC2150
         S      1,EIGHTY                                               EDIC2160
         MVC    0(8,1),WORD                                            EDIC2170
         LM     14,12,12(13)                                           EDIC2180
         MVI    12(13),X'FF'                                           EDIC2190
         MVC    IER+3(1),IER                                           EDIC2200
         MVC    WORD(1),BLANK                                          EDIC2210
         MVC    WORD+1(7),WORD                                         EDIC2220
         MVC    LOWLIM(8),LOWLM1                                       EDIC2230
         MVC    MAX(4),MAX1                                            EDIC2240
         MVC    MIN(4),MIN1                                            EDIC2250
         BCR    15,14                                                  EDIC2260
         DS     0F                                                     EDIC2270
EIGHTY   DC     X'80000000'                                            EDIC2280
FLAG     DC     X'0001'                                                EDIC2290
         DC     X'00'                                                  EDIC2300
TEMP     DC     X'00'                                                  EDIC2310
ADDRESS  DS     F                                                      EDIC2320
         END                                                           EDIC2330
```

APPENDIX I

LISTINGS OF INTEGRATION SUBROUTINES

A. QUADRATIC ELEMENTS - TWO GAUSS POINTS

```
      SUBROUTINE CUB                                                    CU200020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU200030
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD, CU200040
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL             CU200050
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),     CU200060
     1RB3(60)                                                           CU200070
      DIMENSION   STK(60,60),AK(60,60),B(6,60)                          CU200080
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU200090
     1AK3(1,1))                                                         CU200100
      DIMENSION XI(2)                                                   CU200110
      DATA XI/0.57735026918962258,-0.57735026918962258/                 CU200120
C                                                                       CU200130
      DO 100 I=1,NST                                                    CU200140
      DO 100 J=1,NST                                                    CU200150
  100 STK(I,J)=0.000                                                    CU200160
C                                                                       CU200170
C     SELECTION OF POINTS OF INTEGRATION AND FORMATION OF               CU200180
C       INTEGRANT BY SUBROUTINE 'FORMK'                                 CU200190
C                                                                       CU200200
      DO 200 I=1,2                                                      CU200210
      X=XI(I)                                                           CU200220
      DO 200 J=1,2                                                      CU200230
      Y=XI(J)                                                           CU200240
      DO 200 K=1,2                                                      CU200250
      Z=XI(K)                                                           CU200260
C                                                                       CU200270
      CALL FORMK(X,Y,Z,1)                                               CU200280
C                                                                       CU200290
C     INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT          CU200300
C       STIFFNESS MATRIX                                                CU200310
C                                                                       CU200320
      DO 200 L=1,NST                                                    CU200330
      DO 200 M=L,NST                                                    CU200340
  200 STK(L,M)=STK(L,M)+ AK(L,M)                                        CU200350
      DO 300 I=1,NST                                                    CU200360
      DO 300 J=I,NST                                                    CU200370
  300 STK(J,I)=STK(I,J)                                                 CU200380
C                                                                       CU200390
      RETURN                                                            CU200400
      END                                                               CU200410
                                                                        CU200420
```

```
      SUBROUTINE CUB                                                    CU300020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU300040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  CU300050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL             CU300060
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),     CU300070
     1RB3(60)                                                           CU300080
      DIMENSION   STK(60,60),AK(60,60),B(6,60),XI(3),AI(3),AIA(3,3,3)   CU300090
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU300100
     1AK3(1,1))                                                         CU300110
      DATA XI/0.7745966692414834,0.0D0,-0.7745966692414834/             CU300120
      DATA AI/0.5555555555555556,0.8888888888888889,0.5555555555555556/ CU300130
C                                                                       CU300140
      DO 100 I=1,NST                                                    CU300150
      DO 100 J=1,NST                                                    CU300160
  100 STK(I,J)=0.0D0                                                    CU300170
CCC                                                                     CU300180
CCC           COMPUTATION OF PRODUCTS OF WEIGHTS                        CU300190
CCC                                                                     CU300200
      DO 150 I=1,3                                                      CU300210
      DO 150 J=1,3                                                      CU300220
      DO 150 K=1,3                                                      CU300230
  150 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CU300240
CCCC                                                                    CU300250
CCCC          SELECTION OF POINTS OF INTEGRATION AND FORMATION OF       CU300260
CCCC          INTEGRANT BY SUBROUTINE 'FORMK'                           CU300270
CCC                                                                     CU300280
      DO 250 I=1,3                                                      CU300290
      X=XI(I)                                                           CU300300
      DO 250 J=1,3                                                      CU300310
      Y=XI(J)                                                           CU300320
      DO 250 K=1,3                                                      CU300330
      Z=XI(K)                                                           CU300340
      CALL FORMK(X,Y,Z,1)                                               CU300360
C                                                                       CU300370
CCCC          INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT  CU300380
CCCC          STIFFNESS MATRIX                                          CU300390
CCC                                                                     CU300400
      DO 200 L=1,NST                                                    CU300410
      DO 200 M=L,NST                                                    CU300420
  200 STK(L,M)=STK(L,M)+AIA(I,J,K)*AK(L,M)                              CU300430
  250 CONTINUE                                                          CU300440
      DO 300 I=1,NST                                                    CU300450
      DO 300 J=I,NST                                                    CU300460
  300 STK(J,I)=STK(I,J)                                                 CU300470
      RETURN                                                            CU300490
      END                                                               CU300500
```

```
      SUBROUTINE CUB                                                    CU400020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU400040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  CU400050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NPEC7,KEL             CU400060
      COMMON /B3/ AK1(60,60),AK2(60,60),AK3(60,60),RB1(60),RB2(60),     CU400070
     1RB3(60)                                                           CU400080
      DIMENSION   STK(60,60),AK(60,60),B(6,60),XI(4),AI(4),AIA(4,4,4)   CU400090
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU400100
     1AK3(1,1))                                                         CU400110
      DATA XI/0.86113631159405 26, 0.339981043584 8563,                 CU400120
     1       -0.339981043584 8563,-0.86113631159405 26/                 CU400130
      DATA AI/0.347854845137 4539,0.652145154862 5461,                  CU400140
     1        0.652145154862 5461,0.347854845137 4539/                  CU400150
      DO 100 I=1,NST                                                    CU400170
      DO 100 J=1,NST                                                    CU400180
  100 STK(I,J)=0.0D0                                                    CU400190
                                                                        CU400200
C                                                                       CU400210
C             COMPUTATION OF PRODUCTS OF WEIGHTS                        CU400220
C                                                                       CU400230
      DO 150 I=1,4                                                      CU400240
      DO 150 J=1,4                                                      CU400250
      DO 150 K=1,4                                                      CU400260
  150 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CU400270
C                                                                       CU400280
C             SELECTION OF POINTS OF INTEGRATION AND FORMATION OF       CU400290
C             INTEGRANT BY SUBROUTINE 'FORMK'                           CU400300
C                                                                       CU400310
      DO 250 I=1,4                                                      CU400320
      X=XI(I)                                                           CU400330
      DO 250 J=1,4                                                      CU400340
      Y=XI(J)                                                           CU400350
      DO 250 K=1,4                                                      CU400360
      Z=XI(K)                                                           CU400380
      CALL FORMK(X,Y,Z,1)                                               CU400390
C                                                                       CU400400
C             INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT  CU400410
C             STIFFNESS MATRIX                                          CU400420
C                                                                       CU400430
      DO 200 L=1,NST                                                    CU400440
      DO 200 M=L,NST                                                    CU400450
  200 STK(L,M)=STK(L,M)+AIA(I,J,K)*AK(L,M)                              CU400460
  250 CONTINUE                                                          CU400470
      DO 300 I=1,NST                                                    CU400480
      DO 300 J=I,NST                                                    CU400490
  300 STK(J,I)=STK(I,J)                                                 CU400510
      RETURN
```

```
      SUBROUTINE CUB                                                    CU300020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU300040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  CU300050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL             CU300060
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),     CU300070
     1RB3(96)                                                           CU300080
      DIMENSION   STK(96,96),AK(96,96),B(6,96),XI(3),AI(3),AIA(3,3,3)   CU300090
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU300100
     1AK3(1,1))                                                         CU300110
      DATA XI/0.7745966692414834,0.0D0,-0.7745966692414834/             CU300120
      DATA AI/0.5555555555555556,0.8888888888888889,0.5555555555555556/ CU300130
      DO 100 I=1,NST                                                    CU300150
      DO 100 J=1,NST                                                    CU300160
  100 STK(I,J)=0.0D0                                                    CU300170
C                                                                       CU300180
CCC           COMPUTATION OF PRODUCTS OF WEIGHTS                        CU300190
C                                                                       CU300200
      DO 150 I=1,3                                                      CU300210
      DO 150 J=1,3                                                      CU300220
      DO 150 K=1,3                                                      CU300230
  150 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CU300240
C                                                                       CU300250
CCCC          SELECTION OF POINTS OF INTEGRATION AND FORMATION OF       CU300260
              INTEGRANT BY SUBROUTINE 'FORMK'                           CU300270
C                                                                       CU300280
      DO 250 I=1,3                                                      CU300290
      X=XI(I)                                                           CU300300
      DO 250 J=1,3                                                      CU300310
      Y=XI(J)                                                           CU300320
      DO 250 K=1,3                                                      CU300330
      Z=XI(K)                                                           CU300340
C                                                                       CU300350
      CALL FORMK(X,Y,Z,1)                                               CU300360
C                                                                       CU300370
CCCC          INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT  CU300380
              STIFFNESS MATRIX                                          CU300390
C                                                                       CU300400
      DO 200 L=1,NST                                                    CU300410
      DO 200 M=L,NST                                                    CU300420
  200 STK(L,M)=STK(L,M)+AIA(I,J,K)*AK(L,M)                              CU300430
  250 CONTINUE                                                          CU300440
      DO 300 I=1,NST                                                    CU300450
      DO 300 J=I,NST                                                    CU300460
  300 STK(J,I)=STK(I,J)                                                 CU300470
      RETURN                                                            CU300490
      END                                                               CU300500
```

E. CUBIC ELEMENTS - FOUR GAUSS POINTS

```
      SUBROUTINE CUB                                                    CU4000020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU4000040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD, CU4000050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDEP,NLINE,NREC7,KEL             CU4000060
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),PB1(96),RB2(96),     CU4000070
     1PB3(96)                                                           CU4000080
      DIMENSION   STK(96,96),AK(96,96),B(6,96),XI(4),AI(4),AIA(4,4,4)   CU4000090
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU4000100
     1AK3(1,1))                                                         CU4000110
      DATA XI/0.86113631159405260,-0.33998104358485630,                 CU4000120
     1 -0.33998104358485630,0.86113631159405260/                        CU4000130
      DATA AI/0.34785484513745390,0.65214515486254610,                  CU4000140
     1 0.65214515486254610,0.34785484513745390/                         CU4000150
      DO 100 I=1,NST                                                    CU4000160
      DO 100 J=1,NST                                                    CU4000170
  100 STK(I,J)=0.0D0                                                    CU4000180
C                                                                       CU4000190
C     COMPUTATION OF PRODUCTS OF WEIGHTS                                CU4000200
C                                                                       CU4000210
      DO 150 I=1,4                                                      CU4000220
      DO 150 J=1,4                                                      CU4000230
      DO 150 K=1,4                                                      CU4000240
  150 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CU4000250
C                                                                       CU4000260
C     SELECTION OF POINTS OF INTEGRATION AND FORMATION OF               CU4000270
C     INTEGRANT BY SUBROUTINE 'FORMK'                                   CU4000280
C                                                                       CU4000290
      DO 250 I=1,4                                                      CU4000300
      X=XI(I)                                                           CU4000310
      DO 250 J=1,4                                                      CU4000320
      Y=XI(J)                                                           CU4000330
      DO 250 K=1,4                                                      CU4000340
      Z=XI(K)                                                           CU4000350
      CALL FORMK(X,Y,Z,1)                                               CU4000360
C                                                                       CU4000370
C     INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT          CU4000380
C     STIFFNESS MATRIX                                                  CU4000390
C                                                                       CU4000400
      DO 200 L=1,NST                                                    CU4000410
      DO 200 M=L,NST                                                    CU4000420
  200 STK(L,M)=STK(L,M)+AIA(I,J,K)*AK(L,M)                              CU4000430
  250 CONTINUE                                                          CU4000440
      DO 300 I=1,NST                                                    CU4000450
      DO 300 J=I,NST                                                    CU4000460
  300 STK(J,I)=STK(I,J)                                                 CU4000470
      RETURN                                                            CU4000480
      END                                                               CU4000490
                                                                        CU4000510
```

```
      SUBROUTINE CUB                                                    CU500020
      IMPLICIT REAL*8(A-H,O-Z)                                          CU500040
      COMMON /NB1/NEL,NDPT,NPEL,NDF,NEQ,NN,MM,NS,NCOUNT,NST,NSTF,NCLD,  CU500050
     1NPBC,NPBC2,NBCL,NBCH,NMAT,NCON21,NDFP,NLINE,NREC7,KEL             CU500060
      COMMON /B3/ AK1(96,96),AK2(96,96),AK3(96,96),RB1(96),RB2(96),     CU500070
     1RB3(96)                                                           CU500080
      DIMENSION STK(96,96),AK(96,96),B(6,96)                            CU500090
      EQUIVALENCE (STK(1,1),AK1(1,1)),(AK(1,1),AK2(1,1)),(B(1,1),       CU500100
     1AK3(1,1))                                                         CU500110
      DIMENSION XI(5),AI(5),AIA(5,5,5)                                  CU500120
      DATA XI/0.9061798459386640,0.5384693101056831,0.0D0,-0.538469310CU500130
     156831,-0.9061798459386640/                                        CU500140
      DATA AI/0.2369268850561891,0.4786286704993665,0.5688888888888889,0CU500150
     1.4786286704993665,0.2369268850561891/                             CU500160
      DO 100 I=1,NST                                                    CU500180
      DO 100 J=1,NST                                                    CU500190
  100 STK(I,J)=0.0D0                                                    CU500200
C                                                                       CU500210
CCC           COMPUTATION OF PRODUCTS OF WEIGHTS                        CU500220
C                                                                       CU500230
      DO 150 I=1,5                                                      CU500240
      DO 150 J=1,5                                                      CU500250
      DO 150 K=1,5                                                      CU500260
  150 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CU500270
C                                                                       CU500280
CCCC          SELECTION OF POINTS OF INTEGRATION AND FORMATION OF       CU500290
CCCC          INTEGRANT BY SUBROUTINE 'FORMK'                           CU500300
C                                                                       CU500310
      DO 200 I=1,5                                                      CU500320
      X=XI(I)                                                           CU500330
      DO 200 J=1,5                                                      CU500340
      Y=XI(J)                                                           CU500350
      DO 200 K=1,5                                                      CU500360
      Z=XI(K)                                                           CU500370
      CALL FORMK(X,Y,Z,1)                                               CU500390
C                                                                       CU500400
CCCC          INTEGRATION BY GAUSSIAN FORMULA AND FORMATION OF ELEMENT  CU500410
CCCC          STIFFNESS MATRIX                                          CU500420
C                                                                       CU500430
      DO 200 L=1,NST                                                    CU500440
      DO 200 M=L,NST                                                    CU500450
  200 STK(L,M)=STK(L,M)+AIA(I,J,K)*AK(L,M)                              CU500460
      DO 300 I=1,NST                                                    CU500470
      DO 300 J=I,NST                                                    CU500480
  300 STK(J,I)=STK(I,J)                                                 CU500490
      RETURN                                                            CU500510
```

LIST OF REFERENCES

1. Cantin, G., "Three Dimensional Finite Element Studies," (report in preparation).

2. Cantin, G., "An Equation Solver of Very Large Capacity," International Journal for Numerical Methods in Engineering, v. 3, p. 379-388, 1971.

3. Fox, L., An Introduction to Numerical Linear Algebra, Oxford University Press, 1965.

4. Holand, I. and others, Finite Element Methods in Stress Analysis, Tapir, 1969.

5. Irons, B. M., "A Frontal Solution Program for Finite Element Analysis," International Journal for Numberical Methods in Engineering, v. 2, p. 5-32, 1970.

6. Przemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill, 1968.

7. Washizu, K., Variational Methods in Elasticity and Plasticity, ch. 3, Pergamon Press, 1968.

8. Zienkewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Naval Weapons Center c/o Dr. Arnold Adicoff China Lake, California	2
4.	Professor Gilles Cantin, Code 59 Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Professor Robert E. Newton, Code 59 Ne Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Asst Professor David Salinas, Code 59 Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Professor J. E. Brock, Code 59 Bc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Professor R. H. Nunn, Code 59 Nu Chairman, Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	LCDR Emmanuel Leonidas, HN Hellenic Navy Command Athens, Greece	2

		No. Copies
10.	Professor O. C. Zienkewicz Department of Civil Engineering University of Wales Swansea, England	1
11.	Professor T. H. Pian Massachusetts Institute of Technology Boston, Massachusetts	1
12.	Professor Ray W. Clough University of California Berkeley, California	1
13.	Professor Edward L. Wilson University of California Berkeley, California	1
14.	Professor David N. Murray University of Alberta Edmonton, Alberta, Canada	1
15.	LT C. Pfeifer, USN Naval Destroyer School U.S. Naval Base Newport, Rhode Island	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A GENERAL PURPOSE THREE DIMENSIONAL STRESS ANALYSIS PROGRAM

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1971

5. AUTHOR(S) (First name, middle initial, last name)

Emmanuel Leonidas
Lieutenant Commander, Hellenic Navy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	173	8

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

A computerized solution for three dimensional stress analysis was developed. The solution was based on the finite element technique employing twenty and thirty-two nodal point three dimensional isoparametric elements. The solution is applicable to linear elastic structures composed of homogeneous and isotropic materials subjected to static loading under constant temperature. Deformed states are restricted to small displacements and displacement gradients. The program is coded in standard FORTRAN IV and is machine independent except for one subroutine which may be adjusted to specific installations. Independence of core space requirements is achieved by use of direct access storage facilities.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
finite element						
three dimensional stress analysis						

Thesis
L543 Leonidas
c.1 A general purpose
three dimensional str-
ess analysis program.
133154